Two-dimensional Numerical Simulation of Multiphase Flows in MATLAB

Saadi Daftari

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

October 2010

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the Thesis prepared,

By: **Saadi Daftari**

Entitled: **"Two-dimensional Numerical Simulation of Multiphase Flows in MATLAB"**

and submitted in partial fulfillment of the requirements for the Degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

Dr. Z.C. Chen	Chair
Dr. H.D. Ng	Examiner
Dr. S. Li Building, Civil and Environmental Engineering	Examiner External
Dr. G. Vatistas	Supervisor
Dr. H.A. Abderrahmane	Co-Supervisor

Approved by:

Dr. A.K.W. Ahmed, MASc Program Director
Department of Mechanical and Industrial Engineering

Dean Robin Drew
Faculty of Engineering & Computer Science

Date:

Abstract

Two-dimensional Numerical Simulation of Multiphase Flows in MATLAB

Saadi Daftari

A robust and efficient numerical method for the simulation of incompressible, immiscible, two-phase flows in two dimensions is presented. Following a comprehensive literature review, the one-fluid model is selected to account for the discontinuities in material properties across the interface. The model uses the volume-of-fluid (VOF) method and the continuum surface force (CSF) model to track the interface and account for the surface tension forces, respectively. Integration in time is accomplished explicitly through the forward Euler method and the Navier-Stokes equations are discretized in space using a second-order upwind biased oscillation free Total Variation Diminishing (TVD) method for the convective terms and a second-order central differencing method for the viscous terms. The velocity-pressure coupling is done using a two step first-order projection method. The interface is reconstructed geometrically using a piecewise linear interface calculation (PLIC) method and the VOF advection equation is advanced in time using a directional split algorithm. The final numerical algorithm is implemented in MATLAB where the Pressure Poisson Equation (PPE) obtained from the projection

method is solved efficiently in a cluster environment using a built-in MATLAB solver. Various benchmark tests are used to validate the algorithm and a grid refinement test is performed to determine its formal order of accuracy. Applicability to practical physical and engineering problems is verified over a wide range of test conditions by using the numerical method for the simulation of the rise of a bubble in a liquid and the falling liquid film flow. For both cases the results compare well with the past experimental and numerical results; the model successfully predicts the shape of the rising bubble and captures the complex wave dynamics of the falling liquid film. In particular, the results are on a par with the numerical results obtained using free-surface flow codes in which the dynamics of the gas phase are ignored in favour of robustness and stability.

Using MATLAB to implement and run the final code proves to be advantageous as the native parallel computing capabilities of MATLAB reduces the computational time of the simulations significantly while the data presentation and visualization is facilitated using the built-in tools that MATLAB provides. The final implementation of the algorithm involves less than 200 lines of code which makes it easy for the code to be maintained and improved.

Acknowledgements

Firstly, I would like to thank my supervisors Dr. Georgios H. Vatistas and Dr. Hamid A. Abderrahmane for their continuous support and in particular for their patience throughout my graduate years. The completion of this thesis was an unpredictable and challenging journey that helped me grow on an academic and personal level. Your helpful advice and encouragements are greatly appreciated.

Also, I would like to thank Mr. Joel Krajden for providing me with the scripts needed to set up and run the code on the HPC cluster at Concordia University.

Last but not least, I would like to thank my dear parents for their continuous love and support that gave me the energy and motivation to complete this thesis which at times seemed to be an impossible task and Nogol Madani who was the one friend that I needed to help me get through these times with a smile on my face.

Contents

List of Figures

List of Tables

Nomenclature

A	area, m^2
Ar	Archimedes number
Bo	Bond number
C	characteristic function
Ca	capillary number
C_{CFL}	Courant number
d	diameter of the bubble, m
E	error
$\hat{e}$	unit gravity vector
F	volume fraction
$\bar{F}$	finite Laplacian filter applied to the F field
$\tilde{I}$	smoothed volume fraction
$\vec{F}_b$	body force, N/m^3
Fr	Froude number
$\vec{F}_{sa}$	surface tension force vector per unit interfacial area, N/m^2
$\vec{F}_{sv}$	surface tension volume force vector, N/m^3
f_p	forcing frequency, Hz

$\vec{g}$	gravitational force vector, m/s^2
h_c	initial flat film thickness
I	3x3 unit tensor
$\hat{i}, \hat{j}$	unit vector in the x and y direction
k	local mean curvature, 1/m
M	Morton number
m_f	number of filter application to the F field
N_{mesh}	number of mesh cells
N	number of time steps
$\vec{n}$	inward normal vector, 1/m
$\hat{n}$	unit normal to the interface
P	scalar pressure, N/m^2
P_s	pressure jump across the interface
$\overline{p}$	truncation rate decay
Re	Reynolds number
r	ratio of upwind-side gradient to downwind-side gradient
rr	mesh refinement ration
S	solution of the coarser mesh
$\overline{S}$	solution of the finer mesh
sb	bottom side fraction of the cell lying within the fluid

sl	left side fraction of the cell lying within the fluid
sr	right side fraction of the cell lying within the fluid
st	top side fraction of the cell lying within the fluid
T	stress tenser
t	time, s
Δt	time step size, s
U_∞	measured bubble terminal rising velocity, m/s
u,v	velocity component in the x and y directions, m/s
u_c	average velocity of the fully developed flow, m/s
u_{in}	normalized inlet velocity profile with a time dependant perturbation
u_p	monochromatic time varying inlet perturbation
u_w	normalized wave velocity
$\vec{V}$	velocity vector, m/s
We	Weber number
$\vec{X}$	position vector of point x within the flow domain, m
x,y	spatial coordinates in 2D
$\Delta x, \Delta y$	computational cell dimension in the x and y directions, m

Acronyms

CLSVOF	coupled level-set volume-of-fluid

CSF	continuum surface force
CST	continuum surface tension
CV	control volume
FD	finite difference
FV	finite volume
FLIP	fluid implicit particle
LBM	lattice Boltzmann method
MAC	marker-and-cell
PIC	particle-in-cell
PLIC	piecewise linear interface calculation
PPE	pressure Poisson equation
SLIC	simple line interface calculation
SPH	smoothed particle hydrodynamics
TVD	total variation diminishing
VOF	volume-of-fluid

Greek Letters

α	angle the interface makes with the positive x axis within the 1x1 cell, °
β	angle the interface makes with the positive x axis within the cell, °
γ	inclination angle of the plate, °
δ	Dirac delta function

ε	dimensionless disturbance magnitude
λ	dimensionless wavelength
μ	dynamic viscosity, N.s/m^2
ν	kinematic viscosity, m^2/s
ρ	density, kg/m^3
σ	surface tension, N/m
ϕ	general scalar quantity
ψ	TVD limiter function

Subscripts

1	fluid 1
2	fluid 2
g	gas phase
I	interface
i,j	mesh indices in the x and y directions
L	liquid phase
l	left cell edge
r	right cell edge
s	south wall
x	component in the x coordinate
y	component in the y coordinate

Superscripts

*	intermediate parameter
~	normalized parameter
n	current time level
n+1	advanced time level

Chapter 1

Introduction

1.1 Background

Interfacial multiphase flows involving two or more distinct phases separated with a well defined interface occur in a variety of physical and technological processes such as condensers, evaporators, chemical reactors, casting processes, mold filling, extrusion and spray decomposition (Puckett, Almgren, Bell, Marcus, & Rider, 1996; Kunugi & Kino, 2005). Natural and biological phenomena such as ocean wave and blood flow are also multiphase in nature and highlight the broad spectrum of physical problems where these flows are encountered (Yeoh & Tu, 2010).

Multiphase flows form a rather challenging class of fluid dynamics problems due to the moving fluid interfaces within the domain of the flow, discontinuity in material properties and a complicated flow field near the interface (Yeoh & Tu, 2010). These complexities greatly limit the effectiveness of purely analytical studies and necessitate special setup for the experimental studies since many of the techniques developed for single-phase flows are inadequate for multiphase flows (Prosperetti & Tryggvason, 2007).

Numerical simulation represents an economical and efficient means of investigating complex fluid dynamics problems, circumventing the inherent difficulties and limitations

associated with experimental studies (Bierbrauer, 2004). It is, therefore, an effective alternative to tackle multiphase flow problems and reveal details of flow physics that otherwise could not be studied experimentally or analytically (Yeoh & Tu, 2010). The flexibility of numerical simulation allows for the study of multiphase flows to be extended over a wide range of fundamental physical parameters such as gravity, surface tension, density and viscosity ratios (Prosperetti & Tryggvason, 2007).

1.2 Objectives and Scope

This work is motivated by the numerous advantages that the numerical simulation of multiphase flows offers and aims to provide a robust and efficient method for the simulation of a broad range of multiphase flow problems. Successful application to the falling thin film flow is of prime interest due to its many industrial applications and the challenges associated with its numerical simulation. Also, the numerical scheme is to be used as a baseline for future developments and therefore needs to be clear in concept and easy to implement and use. Moreover, the final numerical scheme is implemented in MATLAB to take advantage of its parallel computing capabilities and simplify the code implementation and data post processing by using the large library of built-in functions that MATLAB provides to manipulate and solve the large system of equations associated with numerical simulations.

Although the code developed in this thesis can be extended and applied to many problems involving multiphase flows, the discussion herein is restricted to the case of incompressible, immiscible, two-phase flows in a two-dimensional domain. The code

will not include any turbulent modeling or energy equation (only isothermal flows are considered).

1.3 Organization

Chapter two starts by presenting a detailed discussion on some of the common numerical models for multiphase flow problems, their underlying principles and their respective governing equations. A review of some of the techniques employed to track the location and the shape of the interface in multiphase flows is presented and their respective advantages and disadvantages are highlighted. The results of the review are then used to select the mathematical model that is most suitable for the purpose of this thesis.

In chapter three a detailed description of the discretization techniques used to approximate the governing equations is presented. Justifications are offered for selecting MATLAB as the programming environment in regards to the implementation and performance of the final code.

Chapter four starts by performing relevant benchmark tests on individual components of the numerical scheme to assess the performance of each scheme and better understand their characteristics. This is followed by a grid refinement test in order to quantify the accuracy of the overall scheme and identify the areas that need improvements.

In chapter five applications to the rising bubble and the falling thin film flow are presented. Comparison to the results of the past experiments and other numerical

simulations is used to validate the numerical scheme and measure its capabilities and limitation for the simulation of practical multiphase flows.

Finally, the work is concluded by reiterating the objectives of this thesis followed by an assessment of the overall performance of the numerical scheme and the future improvements required to better meet the initial objectives.

Chapter 2

Review of Mathematical Models for Multiphase Flows

2.1 Overview

Modeling the motion of an interfacial multiphase flow must account not only for the usual flow parameters, such as the velocity and pressure, but should also consider the discontinuity of the material properties such as density and viscosity across the flow interface, the position of the moving and deforming interface between the phases and the interfacial phenomena such as surface tension at the phase interface (Yeoh & Tu, 2010). In most cases, the mathematical models developed for single-phase flows also hold for the motion of multiphase flows with additional considerations needed to account for the abrupt changes in material property and interfacial effects (Tryggvason, Sussman, & Hussaini, 2007).

The two approaches commonly used for modeling of multiphase flows are the sharp-interface model and the immersed interface model. In the former approach the governing equations are solved for each phase separately and the appropriate interfacial boundary conditions are used to couple the fluid phases. Thus the main challenge for the sharp interface model is to maintain an explicit definition of the interface to which the interfacial boundary conditions are applied. This approach often involves meshes that conform to the shape of the interface or use Lagrangian particle

methods to maintain a sharp definition of the interface. In the latter approach the entire flow domain is considered as a whole with different fluids accounted for by introducing variable flow properties across the domain. Here, the interfacial effects are considered by adding appropriate terms to the governing equations. The interface in this case could be identified as the location of abrupt changes in material properties and as such a method to implicitly infer the position and shape of the interface using the distribution of the material properties within the flow domain is required. Many such methods have been developed in the last three decades with successful applications to many industrial and scientific problems. This section aims to offer a detailed review of these approaches, their main contributors and their respective advantages and disadvantages. The equations developed herein can be regarded as a starting point for the numerical prediction of multiphase flows.

2.2 The Sharp Interface Model

The sharp-interface model, also referred to as the jump-condition formulation of the governing equations, consists of decomposing the problem into any number of bulk-phase domains and solving a separate set of governing equations for each phase. The interface between the fluids is then treated as a boundary to which the appropriate interfacial boundary conditions are explicitly applied (Scardovelli & Zaleski, 1999).

Therefore, considering the incompressible flow of two immiscible fluids and assuming constant material properties, the governing equations for each phase of an interfacial flow are the conservation of mass or the continuity equation

$$\nabla \cdot \vec{V} = 0, \tag{2.1}$$

which implies that for an incompressible flow volume is conserved, and the conservation of momentum transport

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\nabla \cdot T + \frac{1}{\rho} F_b, \tag{2.2}$$

where ρ, $\vec{V}(u,v)$, T and F_b are the fluid density, the flow velocity vector defined in the two-dimensional space coordinate, the stress tensor of the phase and a body force, usually the gravitational forces, respectively. Here, u and v are the velocity vector components in the x and y coordinates, respectively. The stress tensor may be decomposed into a pressure term and a viscous term (Prosperetti & Tryggvason, 2007) such that

$$T = -pI + 2\mu D, \tag{2.3}$$

where p is the scalar pressure, μ is the coefficient of dynamic viscosity and I is the 3x3 unit tensor. For a Newtonian fluid the viscous stresses are proportional to the rates of deformation, (Versteeg & Malalasekra, 2007) yielding the following stress tensor formulation

$$D = \frac{1}{2}\left((\nabla \vec{V}) + (\nabla \vec{V})^{T} \right). \tag{2.4}$$

Combining equations (2.2) to (2.4), results in the familiar form of the incompressible Navier-Stokes equation for the momentum transport of a Newtonian flow:

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu\left[\left(\nabla \vec{V}\right) + \left(\nabla \vec{V}\right)^{T}\right] + \vec{g}. \tag{2.5}$$

Here, F_b has been replaced by the gravitational force $\rho\vec{g}$, where $\vec{g}(g_x, g_y)$ is the two-dimensional gravitational vector. Although in their integral form the Navier-Stokes equations allow for arbitrary change in material properties of the fluids, in their differential form they are only applicable to single-phase flows with no discontinuities (Tryggvason et al., 2007). Therefore, for multiphase flows they require the accompanying boundary conditions at the phase interface in order to account for discontinuities and interfacial effects.

The boundary conditions to be applied at the interface can be derived by applying the conservation laws to the interface where conservation of momentum and mass must hold as well (Ferziger & Peric, 2002). Neglecting any mass flux taking place at the interface, whereby the immiscibility of the two interfaces is enforced, the conservation of mass at the interface yields

$$\rho_1\left(\vec{V}_1 - \vec{V}_I\right)\cdot \vec{n} = \rho_2\left(\vec{V}_2 - \vec{V}_I\right)\cdot \vec{n} = 0, \tag{2.6}$$

where subscript 1 and 2 refer to phase 1 and phase 2, and $\vec{V}_I$ is the velocity of the interface. Known as the kinematic condition at the interface, equation (2.6) states that the normal component of the velocity of the fluid phases at the interface are equal to the normal component of the velocity of the interface. This implies that mass flux across

the interface is zero and that the interface is a sharp boundary separating the two fluids (Ferziger & Peric, 2002).

Conservation of momentum at the interface, or the so-called dynamic condition, requires forces at the interface to be at equilibrium (Ferziger & Peric, 2002). That is

$$\left(pI + 2\mu D\right)_1 \cdot \hat{n} - \left(pI + 2\mu D\right)_2 \cdot \hat{n} = \left(\sigma k - \nabla_S \sigma\right) \cdot \hat{n}, \tag{2.7}$$

where $\hat{n}$ is the unit normal to the interface, σ is the surface tension, ∇_S is the surface gradient, and k is the local mean curvature of the interface defined as

$$k = \nabla \cdot \hat{n}. \tag{2.8}$$

It is often more convenient to decompose equation (2.7) into its normal and tangential components. The normal component is

$$p_1 - p_2 = \hat{n} \cdot \left(\mu_1 D_1 - \mu_2 D_2\right) \cdot \hat{n} + k\sigma, \tag{2.9}$$

while the tangential component is

$$\nabla_S \sigma = \hat{n} \times \left(\mu_1 D_1 - \mu_2 D_2\right) \cdot \hat{n}. \tag{2.10}$$

Equation (2.9) implies that the normal force tends to smooth the interface by damping regions of high curvature, whereas the tangential forces defined in equation (2.10) tend to drive the fluid along the interface to the regions of larger σ (Begum & Abdul Basit, 2008). The latter phenomenon is known as the Marangoni or capillary convection (Ferziger & Peric, 2002).

For free-surface flows where the interface separates a liquid from a gas or a vapour, the dynamic effects of the gas phase can be modeled in terms of pressure alone (Prosperetti & Tryggvason, 2007). Thus, neglecting the viscous forces at the interface and assuming σ being constant, as is the case when there is no variation in temperature, surface cleanliness and chemical composition of the flow, the tangential component of the surface tension vanishes and equation (2.7) is reduced to a pressure jump across the interface

$$p_s \equiv p_1 - p_2 = \sigma k. \tag{2.11}$$

Thus, assuming constant surface tension coefficient and neglecting the dynamics of the gas phase, the jump across the free-surface is dictated by the surface curvature variation only.

Equations (2.1) and (2.5), as well as the associated interfacial boundary conditions defined by equations (2.6) and (2.7) , constitute the sharp-interface formulation of the incompressible multiphase flows. This approach often invokes methods that need to track the interface location explicitly and is therefore prone to the difficulties associated with these methods. Some of these methods are briefly discussed in the following sections.

2.3 The Immersed Interface Model

Another approach to account for the discontinuities in material properties and the interfacial phenomena is the immersed interface model, or the so called one-fluid formulation, in which both phases are treated as a single fluid whose properties vary in

space and time (Ferziger & Peric, 2002). In principle, this approach allows for multiphase flows to be treated in the same manner as homogeneous flows using the many available algorithms available for such flows.

In the one-fluid model a single set of the governing equations is solved for the whole flow domain and the interface is not treated as a boundary and hence does not need any boundary conditions to be prescribed on it (Ferziger & Peric, 2002). In this case, the interface is identified as the location where the fluid properties change abruptly. This requires the fluid density and viscosity in equation (2.1) and (2.5) to be defined as a function of material properties, position in space, and time. That is

$$\rho = f\left(\rho_1, \rho_2, x, y, t\right), \tag{2.12}$$

$$\mu = f\left(\mu_1, \mu_2, x, y, t\right). \tag{2.13}$$

Therefore, the difference between the formulation of the Navier-Stokes equation for a single-phase flow and the one-fluid model for multiphase flows lies in the definition of the fluid properties, ρ and μ, expressed as a function of space and time for the latter and as constants for the former.

Even though, unlike the jump-condition formulation, the one-fluid model does not require any boundary conditions to be imposed on the interface, the interfacial phenomena, such as surface tension, need to be accounted for by adding the appropriate interface terms to the governing equations. Since such terms are concentrated at the interface only, the δ-function needs to be employed to represent

them (Tryggvason et al., 2007). Therefore, for the one-fluid model the conservation of momentum, equation (2.5), can be re-expressed as follows

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu\left[\left(\nabla \vec{V}\right) + \left(\nabla \vec{V}\right)^{T}\right] + \vec{g} + \frac{1}{\rho}\kappa\sigma\delta(n)\hat{n}, \qquad (2.14)$$

where $\delta(n)$ is a Dirac delta function acting in the direction normal to the interface. With the singular term added to account for surface tension, this equation is valid for the whole flow domain, including in regions where discontinuities in flow parameters are expected due to the presence of the interface.

Further discussion on the definition of ρ and μ, as well as the modeling of the surface terms in the one-fluid model require the position of the interface to be known and are therefore deferred until a method to track the interface is established.

Equations (2.1) and (2.14) accompanied with the space and time varying material properties defined for the whole domain, equations (2.12) and (2.13), constitute the one-fluid model for the multiphase flows. As is evident by the last term of equation (2.14), to define the curvature of the interface and account for surface tension forces, the shape of the interface needs to be known. Thus, similar to the sharp interface model, in the one-fluid model one needs to employ a method to track the location and the shape of the interface in time. The flow governing equations are coupled with the selected interface tracking method through the local curvature of the interface and the definition of material properties as a function of interface location. The main advantage of the one-fluid model is the fact that it allows for the Eulerian approaches to be applied

to the multiphase flows which in general are easy to implement and are quite robust compared to Lagrangian methods in dealing with interface topological changes (D. B. Kothe, 1999).

2.4 Methods of Tracking the Interface

As previously stated, in evaluating interfacial multiphase flows, one has to pay special attention to the topological changes that occur near the interface between the two phases. Detailed knowledge of interface location and shape is essential to accurately impose the interfacial boundary conditions in the sharp interface model, and to define the material properties and the surface forces as a function of interface location and shape in the one-fluid model. In addition, the location and the shape of the interface are often amongst the most important pieces of information that one needs to obtain from an interfacial multiphase problem solution.

Classical numerical methods designed for smooth solutions perform poorly or do not work at all when employed for the evaluation of multiphase problems with non-smooth or discontinuous solutions across the moving boundaries (Loubenets, 2007). This can be attributed to the smoothing of all variations in flow quantities and the subsequent smearing of discontinuities such as the phase interfaces (Nichols, Hirt, & Hotchkiss, 1980). As a result, over the years several approaches have been developed to track the interface while maintaining its sharp definition within the flow. Without any loss of generality, these methods can be categorized as Eulerian or Lagrangian.

From an interface modeling point of view, multiphase numerical schemes can be categorized as either interface tracking or interface capturing. In interface tracking

methods the interface is tracked explicitly either by marking it with special particles or by attaching it to a mesh surface moving with the interface (Yeoh & Tu, 2010). These methods are Lagrangian in nature, whereby the position history of discrete points $\vec{X}_I$ lying on the interface is given by

$$\frac{d\vec{X}_I}{dt} = \vec{V}_I, \tag{2.15}$$

where $\vec{V}_I$ is the velocity at which the interface point x_I moves. Surface marker, front tracking, moving mesh, and particle schemes are examples of this method.

Interface capturing methods make use of either particles with negligible mass or a marker function to mark the fluids on either side of the interface and "capture" the interface implicitly (Yeoh & Tu, 2010). The marker function, or the so-called characteristic function C, is a discontinuous Heaviside function in the limit of zero mesh spacing which for two-phase flows is defined as

$$C = \begin{cases} C_1 & \text{in fluid 1} \\ C_2 & \text{in fluid 2} \\ > C_1, < C_2 & \text{at the interface,} \end{cases} \tag{2.16}$$

where it is assumed that $C_2 > C_1$ (D. B. Kothe, 1999). In practice, C could have a smooth transition zone between one value and the other (Tryggvason et al., 2007). A point on the interface maintains its position in time which invokes the following Lagrangian invariant statement for the evolution of C (D. B. Kothe, 1999; Tryggvason et al., 2007)

$$\frac{\partial C}{\partial t} + \left(\vec{V}_I \cdot \nabla\right) C = 0. \tag{2.17}$$

Markers and cell (MAC), volume-of-fluid (VOF), level set, phase field and shock capturing methods belong to the interface capturing category. MAC and VOF also belong to a sub-category of interface capturing methods known as volume tracking methods (Salih & Ghosh Moulic, 2009). Some of these methods and their underlying principles, along with their main advantages and disadvantages are reviewed in the subsequent sections.

2.4.1 Eulerian Methods

In the context of modeling the shape and the movement of the interface, Eulerian approaches, in which the grid that is used to solve the Navier-Stokes equation is entirely or quasi-entirely fixed (Scardovelli & Zaleski, 1999), can handle topological changes with ease and resolve interface merges and fragmentations automatically (Shyy, Francois, & Udaykumar, 2001). In order to extract detailed information regarding the interface, Eulerian approaches may need elaborate procedures to infer the interface location (Shyy et al., 2001). Nonetheless, Eulerian methods have enjoyed immense popularity from an academic and industrial point of view, as is evident by the numerous schemes available, and have been used to tackle a wide variety of multiphase flow problems.

2.4.1.1 Surface-Fitted Methods

The main idea in surface-fitted methods is to attach a mesh surface to the interface at all times by maintaining a well-defined mesh, whether body-fitted mesh or unstructured mesh, as depicted in Figure 2.1 (Yeoh & Tu, 2010). Each mesh in the respective fluid domains is allowed to move with the fluid and conform to the shape and structure of

the interface thereby tracking the interface automatically and ensuring an explicit sharp interface definition (Yeoh & Tu, 2010). Such a configuration clearly allows the problem to be tackled with the sharp interface model approach discussed in section 2.2. Surface-fitted methods are capable of producing very accurate results for relatively high Reynolds numbers but remain limited to cases with moderate interface distortion and cannot accommodate interfaces that break apart or intersect (Tryggvason & Balachandar, 2007; Yeoh & Tu, 2010). The main difficulty with these methods remains to be the grid generation and the required computational power that increases as a function of geometric complexity (Balachandar, 2007).

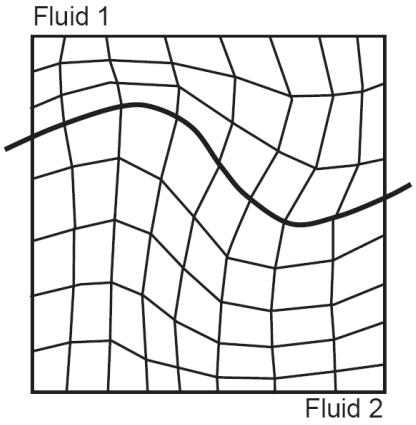

Figure 2.1: Interface representation in the surface fitted method. (Yeoh & Tu, 2010)

2.4.1.2 Surface Marker Method

The basic idea behind this method is to explicitly track an interface by marking it with a set of particles of negligible mass as depicted in Figure 2.2. Originally proposed by Chen et al (1991), Surface Marker (SM) method is essentially a modified version of MAC method in which markers are used only on the interface rather than being distributed throughout the fluid as done in the MAC formulation (Yeoh & Tu, 2010). Due to the

usage of markers only at the interface, this technique is computationally less expensive than the original MAC method (S. Chen, Johnson, & Raad, 1991). The method was further developed into Surface Marker and Micro Cell (SMMC) (S. Chen, Johnson, Raad, & Fadda, 1997) in which smaller (micro) cells are employed near the interface for the discrete representation of the interface and the application of pressure boundary conditions (Yeoh & Tu, 2010). SM method is extremely accurate and makes information such as location, orientation and curvature of the interface explicitly available throughout the whole calculation process (Floryan & Rasmussen, 1989). It also allows for the interface details to be captured on scales much smaller than the Eulerian mesh spacing used (Floryan & Rasmussen, 1989). Similar to MAC method, SM method suffers in situation where the interface stretches or shrinks and needs the addition of fresh markers during the simulation (van Sint Annaland, Deen, & Kuipers, 2005). The method is also sensitive to marker spacing and requires a dynamic redistribution of the surface markers throughout the computation (Yeoh & Tu, 2010). Application to three-dimensional cases is not clear either (Yeoh & Tu, 2010).

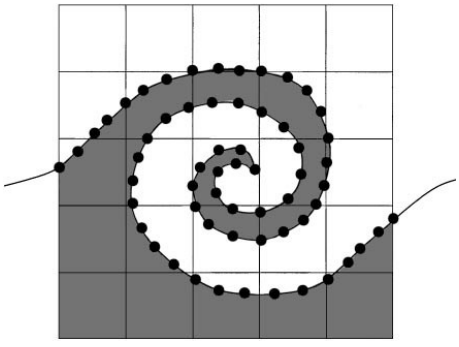

Figure 2.2: An example of the use of surface markers on a stationary Eulerian grid. (Scardovelli & Zaleski, 1999)

2.4.1.3 Front Tracking Method

Front-tracking is best described as a combination of interface tracking and interface capturing where Lagrangian methods are used to treat the interface in an otherwise Eulerian mesh (Tryggvason, Bunner, Ebrat, & Tauber, 1998). Its first application to immiscible multiphase flows appears to be carried out by Unverdi and Tryggvason (1992) and is briefly described herein. As depicted in Figure 2.3, in front tracking method a stationary regular grid is used for the fluid flow, but the interface is explicitly tracked by a separate grid of lower dimension, called the front (Tryggvason et al., 1998). Similar to interface capturing methods, a single set of governing equations is solved on the fixed Eulerian grid for all the phases (Tryggvason et al., 1998). To accommodate for discontinuities across the interface, the interface is purposefully given a finite thickness of the order of mesh size across which fluid properties smoothly transition from one phase to the other (Yeoh & Tu, 2010). Thus, to identify where each fluid is, it is necessary to construct a marker function from the location of the front and then use this marker function to determine the fluid properties near the interface. The calculated fluid properties along with other quantities evaluated at the front, such as surface tension, are then passed to the fixed Eulerian grid through a smoothing function for the calculation of the fluid flow (Tryggvason et al., 2007).

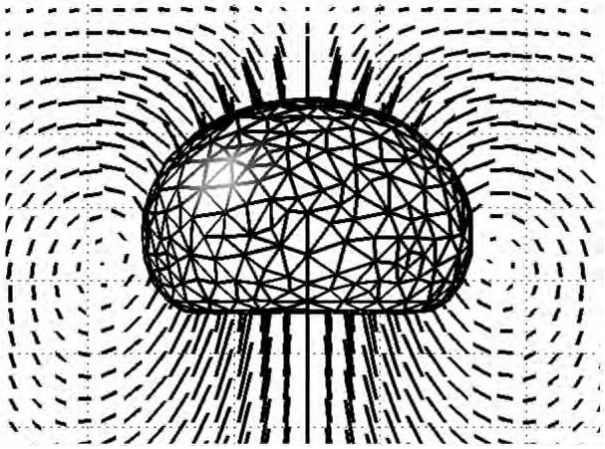

Figure 2.3: Explicit tracking of a three-dimensional fluid interface in front tracking. (Tryggvason et al., 2007)

Front-tracking method is considered to be very accurate with no numerical diffusion and gives the precise location of the interface throughout the numerical calculation which in turn alleviates the evaluation of surface curvature and surface tension forces (Yeoh & Tu, 2010). It also allows for the presence of more than one interface in a computational cell which could be important in some applications (Loubenets, 2007). Arbitrary topological changes are accommodated by adding local topological procedures that would allow interfaces reconnect or break (Bierbrauer, 2004). The major disadvantages of front-tracking method include relative difficulty in handling three-dimensional topological changes and complexity in implementation which mainly arises from the interaction between the Lagrangian front and the Eulerian mesh (Loubenets, 2007). Moreover, the method requires a remeshing of the interface as it becomes distorted and involves solving an elliptical equation for the distribution of the marker function (Rudman, 1997).

2.4.1.4 Marker-and-Cell Method

The marker-and-cell (MAC) method is a volume tracking method that was first introduced by Harlow and Welch (1965) as a variation of particle-in-cell (PIC) method and is considered as the earliest numerical method designed specifically to deal with complicated interfacial flows (Johnson, 1996; Rudman, 1997). The technique had many novel features at the time of its introduction and was the first successful method for incompressible flows (Johnson, 1996). As depicted in Figure 2.4, the MAC method makes use of massless Lagrangian particles to locate the material in the mesh and, consequently, define the location of the surface (Johnson, 1996). A cell with no markers is considered empty while a cell with markers, lying adjacent to an empty cell, is identified as a surface cell. Cells that contain more than one fluid at a time are surface cells as well (Yeoh & Tu, 2010). The evolution of the surface is achieved by moving the markers with the locally interpolated fluid velocity (Salih & Ghosh Moulic, 2009). In MAC method, some special treatment is required to define fluid properties in newly filled cells while values in cells that are emptied need to be reset (Salih & Ghosh Moulic, 2009). Additionally, gas pressure needs to be explicitly assigned at the surface cells while the incompressibility constraint or the zero-shear stress condition is enforced at the interface by extrapolating the velocity components to the cells immediately outside the surface (Yeoh & Tu, 2010).

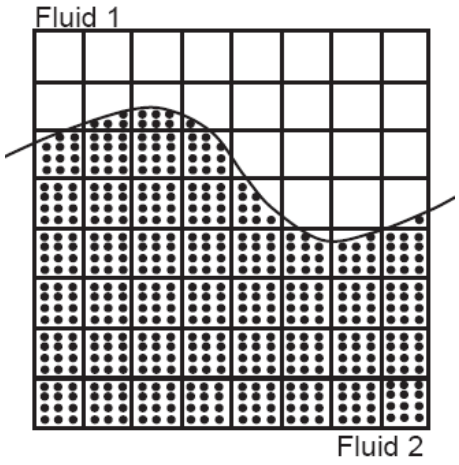

Figure 2.4: MAC representation of an interface using massless particles. (Yeoh & Tu, 2010)

The MAC method can treat any number of fluids and allows for arbitrary interfacial flows subject to large distortions with interacting surfaces (Bierbrauer, 2004). Although extension to three-dimensional cases is fairly simple, the large number of markers needed considerably increases the computational time (Yeoh & Tu, 2010). Also, difficulties arise in the particle representation of the fluids in regions involving converging or diverging flows that could lead to unphysical void region inside the domain (Yeoh & Tu, 2010). Similar to other volume tracking methods, in the MAC method the resolution is limited to the grid size that may result in the smearing of the interface (Bierbrauer, 2004). Finally, the MAC method is unable to determine the orientation of the surface and suffers from instability at the free-surface due to the approximate application of the free-surface boundary conditions (Rudman, 1997).

2.4.1.5 Volume-of-Fluid Method

A popular and successful method to track interfaces in two-phase and free-surface flows, the volume-of-fluid (VOF) method is based on a discrete characteristic function F which is equal to 1 in one phase and to 0 in the other (Scardovelli & Zaleski, 2003; Tryggvason et al., 2007). The discrete characteristic function, also known as the volume fraction function, is define for a two-phase flow as the ratio of the volume of one of the fluids, say fluid 1, to the total volume of the computational cell. Thus, in reference to Figure 2.5, a F value of 1 corresponds to a cell that is entirely filled with the dark fluid while a F value of zero corresponds to a cell that contains no dark fluid and is filled with the light fluid. A value that falls between zero and one implies that the cell contains both fluids and is therefore an interface cell. (Rider & Kothe, 1998; Scardovelli & Zaleski, 1999; Scardovelli & Zaleski, 2003). As with other interface capturing methods, in the VOF method an exact representation of the interface is not retained and the discrete volume fraction field is the only interface information from which the interface geometry can be inferred (D. B. Kothe, Mjolsness, & Torrey, 1991; Rider & Kothe, 1998).

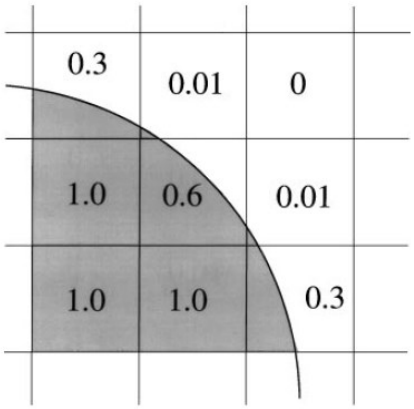

Figure 2.5: The discrete presentation of a smooth circular arc over a square grid using the VOF volume fraction data. (Scardovelli & Zaleski, 1999)

Generally speaking, the VOF method consists of two parts: an interface reconstruction algorithm for determining an approximation to the interface from the volume fraction field information and a volume-of-fluid advection algorithm or a transport algorithm for determining the motion of the F field in time (Puckett et al., 1997). For incompressible fluids volume is conserved and therefore the advection of the interface can be modeled by solving a conservation law, such as equation (2.17) re-written for the F function (Scardovelli & Zaleski, 1999).

In the VOF method the final geometry of the reconstructed interface is not unique for a given F field and depends on the interface reconstruction scheme used (Rider & Kothe, 1998). Thus interface reconstruction is a key part of any VOF method. These methods have evolved significantly during the past a few years and can be categorized as simple line interface calculation (SLIC) and piecewise linear interface calculation (PLIC).

The SLIC method, introduced by Noh and Woodward (1976), approximates the interface as piecewise constant, where interface within cells is assumed to be aligned with one of

the logical mesh coordinates and the fluid configuration in that call is assumed to be different for horizontal and vertical movements (Rider & Kothe, 1998; Yeoh & Tu, 2010). SLIC is a direction-split algorithm and only considers neighbouring cells in the flux direction. This results in a different interface reconstruction for each sweep direction (Figure 2.6b) (Rudman, 1997). Hirt and Nichols (1981) proposed a slightly different approach in which the interface orientation remains the same for all sweep directions although it was still forced to remain parallel to one of the coordinates (Figure 2.6c). The orientation of the interface is determined by the normal to the interface which evaluated using the value of the volume fractions in the neighbouring cells (Tryggvason et al., 2007). This approach led to no significant improvements over the original SLIC method (Rudman, 1997).

Unfortunately, all piecewise constant volume tracking methods suffer from a common effect known as flotsam or jetsam which are isolated, submesh-size material bodies that separate from the main material body due to errors introduced by volume tracking algorithm (Rider & Kothe, 1998; Rudman, 1997; Scardovelli & Zaleski, 2003).

A more accurate and popular method for the reconstruction of the interface is the PLIC method which gained popularity in the early 1980's due to the developments done by Young (Youngs, 1982). In this method the interface is reconstructed by a straight line segment in each cell but no constraint is imposed on the orientation of the line (Figure 2.6d). In this case the slope of the line is given by the interface normal with the intercept determined by invoking volume conservation within the cell (Rider & Kothe, 1998). Therefore, in PLIC method the accuracy of the interface reconstruction depends

on the proper approximation to the interface normal (Tryggvason et al., 2007). Many such approximations are discussed and compared in Pilliod and Puckett (2004). An implementation of a VOF-PLIC method is discussed in greater detail in chapter 3.

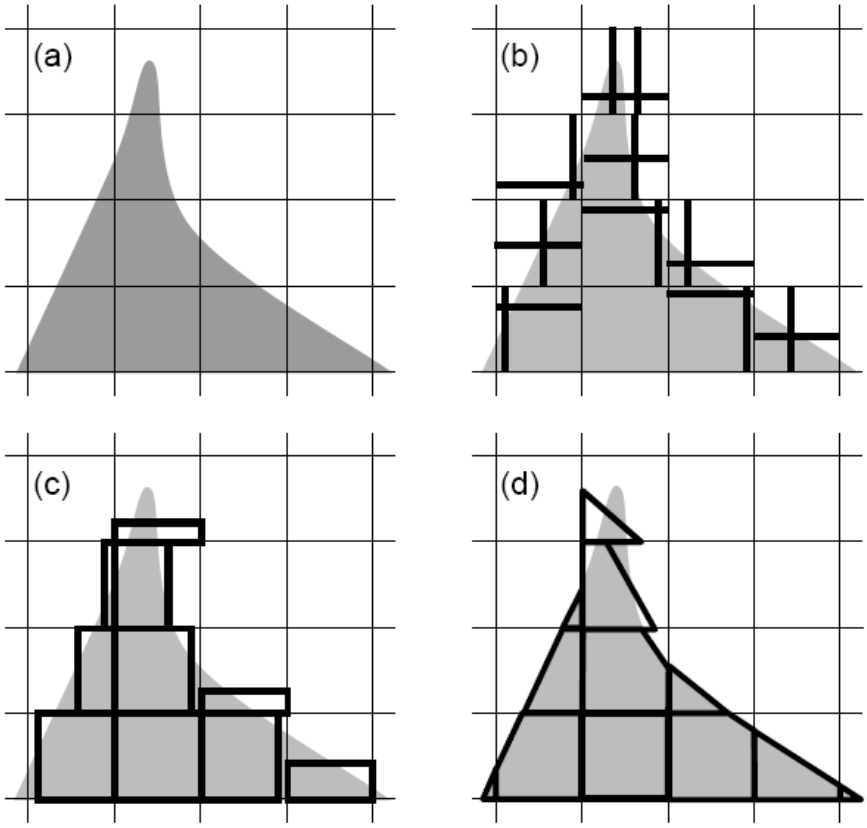

Figure 2.6: The VOF reconstruction of an interface; a) The original interface, b) the SLIC reconstruction, c) the Hirt-Nichols reconstruction, d) the PLIC reconstruction. (Tryggvason et al., 2007)

The VOF method has become the preferred choice when dealing with severe topological changes of interface since it requires no initial assumptions on the nature of the interface and as such no special techniques to accommodate topological evolution of the interface (Bierbrauer, 2004; Puckett et al., 1997). Generally, VOF methods, SLIC or PLIC, are robust and relatively easy to extend from two to three dimensions, preserve

mass inherently and are local in a sense that only volume fraction values in neighbouring cells are needed (Pilliod & Puckett, 2004; Scardovelli & Zaleski, 1999). Drawbacks of VOF methods include the creation of flotsams in SLIC methods, inability to resolve details of interfaces smaller than the mesh, and inability to accurately determine viscous stress and surface tension when curvature and orientation cannot be estimated accurately (Bierbrauer, 2004).

2.4.1.6 Level-Set Method

Level-set method, introduced by Osher and Sethian (1988), is another interface capturing method designed to reduce the numerical diffusion hampering shock-capturing methods (van Sint Annaland et al., 2005). The basic idea is to transfer the discontinuous C function in equation (2.16) into another smooth function ϕ, use equation (2.17) to advect it, and transfer ϕ back to C (D. B. Kothe, 1999). To do so, the interface is represented by the zero contour of a signed distance function, the level-set function ϕ, being positive on one side of the interface and negative on the other side (Olsson & Kreiss, 2005). Unlike the VOF method, where the transition from one fluid to the next takes place over one grid unit, the scalar level-set function transition smoothly across the interface, allowing highly accurate numerical solutions to equations (2.17) solved for ϕ (D. B. Kothe, 1999; Tryggvason et al., 2007). This also leads to convenient formulas for the interface normal and curvature (Puckett et al., 1997). Often the level-set function ceases to be a signed distance function from the interface after it is advected in time (Yeoh & Tu, 2010). Thus, the so called reinitialization of the level-set

function is devised to maintain it as a distance function at all times (Olsson & Kreiss, 2005).

Similar to other interface capturing methods, level-set method can handle topological changes automatically, without any special considerations needed (Puckett et al., 1997) and its extension to three-dimensional flows is straight forward (Tryggvason et al., 2007). For simple flows and especially for flows where the interface moves relatively parallel to one of the coordinates, level-set method yields very accurate results (van Sint Annaland et al., 2005). Unfortunately, in most cases the reinitialization moves the zero level set position, thereby violating mass conservation (D. B. Kothe, 1999). As a results, in flow fields with sever vorticity or in cases where the interface is significantly deformed, level-set method suffers from severe loss of mass and thus loss of accuracy (van Sint Annaland et al., 2005).

2.4.1.7 Phase-Field Method

The phase-field method, also known as the diffused-interface method, is based on modifying the governing equations by incorporating some of the physical effects governing the structure of a thin interface (Tryggvason et al., 2007). Like other interface capturing methods, the phase-field method models the interfacial forces as continuum forces by smoothing discontinuities at the interface over a thin but numerically resolvable layer (D. B. Kothe, 1999). It uses a phase function, φ, which describes the diffused transition between the two fluids (Bierbrauer, 2004). The value of φ is 0 in one fluid and 1 in the other with the interface indentified as regions where φ has a value between 0 and 1 (Bierbrauer, 2004). In this method a continuum surface tension model,

discussed in the subsequent sections, is used to account for surface forces and the phase field is updated by means of a non-linear advection-diffusion formulation known as the Cahn-Hilliard equation (Tryggvason et al., 2007). It is similar to the advection equation (2.17) used for other interface capturing methods with the right hand side being equal to the Laplacian of the chemical potential rather than zero (D. B. Kothe, 1999). As a result, a regular interface width is maintained which can be diffused or ant-diffused keeping the interface compact (Bierbrauer, 2004).

The phase-field model has the advantage of automatically describing topological changes and having a composition field that has a physical meaning both in the bulk phase and in the interface (Loubenets, 2007). It can also easily accommodate complex physics, such as miscible, immiscible and partially miscible phases (Loubenets, 2007). Discretization of the Laplacian can be problematic and could lead to the interface sticking to the mesh (D. B. Kothe, 1999).

2.4.1.8 Shock Capturing Method

In shock capturing method, equation (2.17) is solved directly by applying a continuum advection scheme. In principle, equation (2.17) is a simple form of a hyperbolic equation and as such it can be approximated using any of the traditional numerical methods designed for the solution of such equations (D. B. Kothe, 1999). This approach is quite appealing since such schemes are often an integral part of any flow solver; to treat the convective part of the Navier-Stokes equation, for example (D. B. Kothe, 1999). In principle, continuum advection schemes evaluate the fluxes in equation (2.17) algebraically as opposed to in VOF methods where the fluxes are evaluated

geometrically (D. B. Kothe, 1999). By doing so, the explicit reconstruction of the interface is circumvented which is advantageous in unstructured grids (van Sint Annaland et al., 2005). Unfortunately, although high-resolution schemes, a detailed discussion of which can be found in (LeVeque, 2002), are quite sophisticated, they work less well for discontinuities encountered in multiphase flows and suffer from excess numerical diffusion near the discontinuities (D. B. Kothe, 1999; van Sint Annaland et al., 2005).

2.4.1.9 Hybrid Methods

In hybrid methods the best aspects of different techniques are combined in an attempt to attain an algorithm that is superior to the individual techniques involved. A number of such methods have been developed in the recent years. They include the coupled level-set and VOF methods (CLSVOF) and methods combining level-set or VOF method with markers.

In CLSVOF method, proposed by Sussman and Puckett (2000), the tracking of the interface is accomplished by using the level-set method in which the continuous level-set function is used to evaluate the interface normal and curvature and then is reinitialized to the piecewise linear interface reconstructed from the mass conserving VOF function (Loubenets, 2007; Yeoh & Tu, 2010). This allows for the tracking of the interface and the calculation of its normal and curvature, which are typically difficult in the VOF method, without any loss in mass due to the level-set reinitialization.

In combined marker level-set methods, such as the one proposed by Enright et al. (2002), markers are randomly positioned near the interface, defined by the zero level-set, and are passively advected by the fluid flow. These markers are subsequently employed to re-build the level-set function in under-resolved regions, where the level set method suffers from excessive regularization, thereby improving its volume conservation properties (Yeoh & Tu, 2010).

likewise, in mixed marker and VOF methods, introduced by Aulisa et al. (2003), surface markers are used to define the interface on the grid lines and maintain a smooth interface movement typical of all marker methods with VOF markers added on the local surface inside each cell to enforce mass conservation (Loubenets, 2007).

2.4.2 Lagrangian Methods

Lagrangian approaches in which the coordinate system moves with the fluid provide for an easy prescription of the boundary conditions on the interface but suffer from a loss of accuracy and numerical instability for flows with large surface deformation as a result of a grid that is excessively skewed and unevenly distributed (Shyy et al., 2001). In order to overcome the grid distortion problem a re-meshing strategy could be applied in order to dynamically adjust the grid to prevent entanglement and maintain accuracy (Floryan & Rasmussen, 1989). However, the re-meshing process leads to increased computational costs while the numerical inaccuracy might persist (Floryan & Rasmussen, 1989) .

Since this work is mostly concerned with the Eulerian methods only a brief description of these methods is presented herein. Further readings on these methods can be found in Floryan and Rasmussen (1989), Johnson (1996) and D. B. Kothe (1999).

2.4.2.1 Particle Methods

Particle methods are not strictly developed for multiphase flow applications but present an attractive approach for multiphase flow simulations due to their particle treatment of the fluids whereby the interface can be defined with little or no extra modeling. In many cases, they involve rather novel approaches towards modeling the fluid flows that are beyond the scope of this work and are merely reviewed herein to give a better overview of the various approaches that one can take to model the motion of multiphase flows.

Particle methods are based on concepts adopted from the molecular dynamics (Floryan & Rasmussen, 1989). They are characterized by the use of discrete interacting particles to represent microscopic fluid parcels (Floryan & Rasmussen, 1989; D. B. Kothe, 1999). The state of the fluid is defined by the attributes of the particles, while the evolution of the system is defined by the laws of the interaction of the particles constructed in such a way that the fluid molecular forces are simulated (Floryan & Rasmussen, 1989). By treating the flow domain in such a manner, the non-linear convection terms in the Navier-Stokes equation can simply be modeled as particle motion, rendering the numerical diffusion usually associated with the approximation of these terms to virtually zero (D. B. Kothe, 1999). Moreover, from an interface modeling point of view, in this

approach the particles are specifically associated with a material and thus the interface is automatically tracked (Floryan & Rasmussen, 1989; D. B. Kothe, 1999). Examples of particle methods include the particle-in-cell (PIC) method, the smoothed particle hydrodynamics (SPH) method and the lattice Boltzmann method (LBM). A brief review of these methods is presented in the following sections. Further readings on the particle methods, their history and recent developments can be found in (S. Li & Liu, 2002), and for LBM in particular in (S. Chen, He, & Luo, 2007).

Particle-in-Cell Method

The PIC method proposed by Harlow (1955) is the oldest particle based method and has been used extensively to model highly distorted interfacial flows and served as the basis for the development of the highly popular MAC method. Being a particle-mesh method, PIC employs an Eulerian grid to compute field variables such as pressure and velocity, and simulates material transport from cell to cell in a Lagrangian fashion (Floryan & Rasmussen, 1989). From this perspective PIC can also be categorized as a mixed Lagrangian-Eulerian method whereby a partial elimination of the grid is achieved (Scardovelli & Zaleski, 1999). Despite the Lagrangian treatment of the materials the PIC method suffers from large numerical diffusion due to the transfer of information from the particles to the Eulerian grid and vice-versa (Johnson, 1996). The fluid implicit particle method (FLIP) reduced this numerical diffusion to very low levels by limiting the transfer of information between grid and particles through forcing the particles to carry all relevant fluid information (Johnson, 1996; D. B. Kothe, 1999). PIC methods can conserve mass exactly, treat interfaces subject to a large distortion and can

accommodate flows with interacting interfaces (Bierbrauer, 2004). However, they remain to be computationally very expensive and tend to be susceptible to numerical instability (D. B. Kothe, 1999). Moreover, similar to Eulerian methods, PIC methods suffer from a limited special resolution dictated by the grid size with no additional treatment of the interface within the cells (Floryan & Rasmussen, 1989).

Smoothed Particle Hydrodynamics Methods

Another class of particle methods includes those that require no accompanying mesh where the domain is traversed by particles with or without a direct physical meaning (Scardovelli & Zaleski, 1999). SPH, originated from works by Lucy (1977) and Gingold and Monaghan (1977), belongs to this class of methods and is best described as a general framework for solving differential equations using particles and smoothing kernels that define the intensity of interaction between particles, depending on their mutual distance (Scardovelli & Zaleski, 1999). Thus, SPH can be considered as a discretization tool to solve continuum problems, such as fluid flows governed by the Navier-Stokes equation (S. Li & Liu, 2002). The method takes advantage of the fact that high-order interpolation schemes used to transfer information between particles can also be used to evaluate derivatives and therefore the relevant differential equations (Floryan & Rasmussen, 1989). These methods are more suitable for the simulation of compressible flows and are readily extended to three dimensions in unstructured meshes (D. B. Kothe, 1999). On the other hand, in SPH the treatment of viscous effects is uncertain (Floryan & Rasmussen, 1989) and numerical instability remains an issue (D. B. Kothe, 1999).

Lattice Boltzmann Method

Originally proposed by McNamara and Zanetti (1988) in an effort to reduce the statistical noise in lattice gas automaton, LBM has become a popular numerical scheme for simulation and modeling of fluid flows. A relatively new particle method, LBM is a reduced-order kinetic model conceived to reproduce the Navier-Stokes equation hydrodynamics at a macroscopic level (Kim & Pitch, 2009). It is a reduced-order kinetic model in the sense that instead of solving complicated kinetic equations such as the full Boltzmann equation, LBM deals with the simplified version of this model; i.e. the discrete Boltzmann equation (S. Chen et al., 2007). Due to its mesoscopic nature, in addition to being used as a discretization tool to solve the continuum problems, as is the case with SPH, LBM could also be used to model the underlying physics of the problem (S. Chen et al., 2007; S. Li & Liu, 2002). Most LBM methods developed for multiphase flows can also be categorized as diffused interface models, in that an interface is numerically resolved with a few grid points across it (Kim & Pitch, 2009). Lattice Boltzmann methods developed for multiphase flows have been successfully applied to a wide range of flows and have been particularly successful in flows involving complex geometry and porous media (Kim & Pitch, 2009). Unfortunately, the application of LBM to the multiphase flows has been limited due to numerical instability especially in flows involving fluids with high density ratios (Kim & Pitch, 2009). Moreover, similar to VOF methods, in LBM problems may arise due to interfacial coalescence of the dispersed elements (van Sint Annaland et al., 2005).

2.5 Interfacial Boundary Conditions Models

Surface tension is an inherent characteristic of interfacial flows and as such needs to be accounted for in an interfacial multiphase flow model. This can be done through the proper prescription of the interface boundary conditions, namely the kinematic condition and the dynamic condition defined by equations (2.6) and (2.11) for constant property incompressible flows. Mass conserving interface tracking methods, such as the VOF method, automatically satisfy the kinematic condition at the interface (Gao, Morley, & Dhir, 2003) and require no further treatment of the governing equations in that respect . The application of the dynamic condition, however, is somewhat more challenging due to the dependence of the surface forces on the local mean curvature. Early multiphase numerical algorithms (i.e. Nichols and al. (1980), Hirt and Nichols (1981)) enforced the dynamic condition directly on the reconstructed interface and as such, despite their popularity, suffered from the inaccuracies in estimating the surface tension forces induced by their rough estimate of the interface itself (D. B. Kothe et al., 1991).

Continuum surface tension (CST) models in which the surface tension is interpreted as a continuous, three-dimensional effect across an interface rather than as a boundary condition on the interface (Brackbill, Kothe, & Zemach, 1992), attempt to alleviate the difficulties associated with the explicit application of the interfacial boundary conditions. These methods root back to the work of Peskin (1977) but own their popularity to the largely cited work of Brackbill et al. (1992) wherein the so-called

continuum surface force (CSF) model was formulated. In the CSF model the surface tension is modeled as a body force $\vec{F}_{sv}$ satisfying

$$\lim_{h\to 0}\int_{\Delta V}\vec{F}_{sv}\left(\vec{X}\right)d^{3}x=\int_{\Delta A}\vec{F}_{sa}\left(\vec{X}_{I}\right)dA, \tag{2.18}$$

where $\vec{X}$ is any point within the flow domain, $\vec{X}_I$ is a point on the interface A, $\vec{F}_{sa}$ is the surface tension force per unit interfacial area defined as

$$\vec{F}_{sa}\left(\vec{X}_{I}\right)=\sigma k\left(\vec{X}_{I}\right)n\left(\vec{X}_{I}\right), \tag{2.19}$$

and h is a length in the order of mesh spacing dx (D. B. Kothe et al., 1991). The area integral is over the portion ΔA of the interface lying within the small volume of integration ΔV. It turns out that in methods where the interface is tracked by means of a smooth characteristic function C, such as the one defined for interface capturing methods (equation (2.16)), equation (2.19) can be approximated using the distribution of C within the flow domain as follows

$$\vec{F}_{sv}\left(\vec{X}\right)=\sigma k\left(\vec{X}\right)\nabla C\left(\vec{X}\right), \tag{2.20}$$

where $\hat{n}$ is approximated by ∇C. Here, $\vec{X}$ could be any point within the flow domain, implying that equation (2.20) is applicable throughout the entire domain but is only non-zero near the interface where the gradient of C is not zero.

Therefore, for the one-fluid formulation of the Navier-Stokes equation for the multiphase flows, one way to approximate the surface tension is to replace $\sigma k \delta(n) \hat{n}$ in equation (2.14) with the volume force $\vec{F}_{sv}$ defined by equation (2.20).

2.6 Mathematical Model

The multiphase numerical scheme selected for the purpose of this work needs to be easy to implement, as generally applicable as possible and capable of accommodating higher order approximations in the future. Moreover, it should be capable of handling topological changes on relatively coarse grids thus insuring a relatively low computational cost. Conservation of mass and momentum is essential. The scheme must also be relatively easy to be extended to three dimensions in the future and relatively decoupled from evaluation of other flow field variables such that the manipulation of the interface capturing/tracking portion of the code would require the least amount of change in the rest of the code and vice-versa. Finally, the method of choice is to be used as the basis for future developments and therefore needs to be clear in concept and implementation with an extensive academic background.

Considering these requirements and the results of literature survey presented previously the mathematical model that is most suited for the purpose of this thesis can now be formulated. First, the one-fluid formulation is selected to model the multiphase flows due to its Eulerian nature and ease of implementation. Second, as a result of its integration with the Eulerian approaches, inherent robustness and high accuracy the VOF method is used to track the interface. Finally, to include the interfacial effects the

CSF method is employed due to its simplicity and tight integration with interface capturing methods such as VOF.

Therefore, the motion of the incompressible multiphase flow, with the interfacial effects taken into account via the CSF model can be expressed as

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu\left[\left(\nabla \vec{V}\right) + \left(\nabla \vec{V}\right)^{T}\right] + \vec{g} + \frac{1}{\rho}\sigma\kappa\nabla F, \quad (2.21)$$

$$\nabla \cdot \vec{V} = 0, \quad (2.22)$$

with the VOF representation of the interface defined as

$$\frac{\partial F}{\partial t} + \left(\vec{V} \cdot \nabla\right) F = 0. \quad (2.23)$$

Here, in reference to equation (2.21), the interfacial jump condition, $\sigma k \delta\left(n\right)\hat{n}$, is replaced by its CSF equivalent defined by equation (2.20), where C has been replaced by the VOF characteristic function, F, and $\vec{X}$ omitted for simplicity. The divergence term on the left hand side is referred to as the convective term while the term involving μ on the right hand side is known as the viscous term. Equation (2.23), also known as the VOF advection equation, is the VOF representation of equation (2.17), where $\vec{V}_I$ is replaced by $\vec{V}$ to enforce the kinematic condition at the interface.

As previously stated, the one-fluid formulation requires the fluid density and viscosity to be defined as a function of material properties, space and time (equations (2.12) and (2.13)). The choice of the VOF method to track the interface allows for the

discontinuous material properties across the flow domain to be expressed as a function of $F(x,t)$, giving rise to the following expressions

$$\rho = F\rho_1 + (1-F)\rho_2, \tag{2.24}$$

$$\mu = \frac{\mu_1\mu_2}{\mu_1(1-F)+\mu_2 F}, \tag{2.25}$$

where F is assumed to be equal to 1 in fluid 1 and 0 in fluid 2. Equation (2.25) satisfies the conservation of momentum at the interface better than a simple volume average formulation for the viscosity and is obtained by working with the inverse of viscosity (Tryggvason et al., 1998). Equation (2.21) to (2.25) can be considered as the governing equations for the motion of an incompressible, interfacial multiphase and need no boundary conditions to be prescribed on the fluid interfaces.

2.7 Summary

Review of some of the most common mathematical models for multiphase flows and various approaches for tracking the fluid interface were presented along with their advantages and disadvantages. The one-fluid model and the VOF method were selected as the methods of choice due to their robustness, efficiency, simplicity and the vast spectrum of multiphase problems they are applicable to, including the falling thin film flow. CSF model was selected to include the surface tension effects as a body force in the momentum transport, eliminating the need for an explicit treatment of the interfacial boundary conditions. The final mathematical model is relatively simple and

quite similar to the models used for single-fluid flows therefore making the vast collection of literature available for the approximation of these flows applicable.

Chapter 3

Solution Method and Numerical Approximation

3.1 Overview

In this chapter, an approximate solution to the governing equations outlined in the previous chapter for the multiphase flow problems is presented. An approximate solution is needed since an analytical solution to Navier-Stokes equations can be obtained only in a small number of cases since these equations are non-linear and coupled (Ferziger & Peric, 2002). Also, there exists no explicit governing equation for the pressure field and as such it is necessary to find such an equation for pressure as part of the solution process (Tryggvason & Balachandar, 2007). Thus, any numerical method for the multiphase flow problems must be able to deal with the traditional challenges posed by the solution of the Navier-Stokes and continuity equations in addition to the difficulties associated with numerical approximation of the multiphase flows, i.e. discontinuous fluid properties and the interface shape and location.

The equation set to be solved consists of equations (2.21), (2.22) and (2.23). The numerical scheme presented herein for a solution to these equations involves two coupled tasks. In the first step the flow field velocity and pressure are resolved through a solution to the Navier-Stokes and continuity equation. Here, the fluid properties are computed based on the initial distribution of the VOF function using equations (2.24)

and (2.25). Integration in time is accomplished through a first-order explicit method with the velocity-pressure coupling established by means of a so-called projection method where the pressure field is adjusted so that a divergence free velocity field is obtained (Tryggvason & Balachandar, 2007). Spatial discretization of the convective terms is accomplished by employing an upwind biased total variation diminishing (TVD) scheme to obtain stability near discontinuities. The viscous terms are discretized using the central differencing scheme while a finite difference method is used to approximate the surface tension forces. In the second step, geometric reconstruction of the interface is performed using a PLIC method to determine the amount of fluid fluxed out of each computational cell. Subsequently, given the updated velocity field, the VOF function distribution is advanced in time using a directional split algorithm for the VOF advection equation. These steps are discussed in great details in the ensuing sections.

3.2 Numerical Approximation of the Navier-Stokes Equation

This section aims to present a detailed discussion on the numerical methods used to approximate the solution of the Navier-Stokes equation derived for the one-fluid model. At this stage, it is assumed that the location of the interface is known and that the fluid properties can readily be calculated across the computational domain based on the known volume fraction field. Velocity and pressure are advanced in time upon the completion of this stage.

3.2.1 Integration in Time

The standard way to integrate the Navier-Stokes equation in time is through the so-called projection method whereby the pressure field is implicitly constrained by the continuity equation (Tryggvason & Balachandar, 2007). In this case the pressure takes on whatever value is needed to enforce a divergence-free velocity field.

The projection method consists of two main steps. First, an intermediate velocity field is found without accounting for pressure. Second, the pressure necessary to make the velocity field divergence-free is solved for and is used to correct the intermediate velocity field by adding the pressure gradient (Tryggvason & Balachandar, 2007).

Using a simple forward Euler method to integrate equations (2.21) and (2.22) in time one obtains

$$\frac{\vec{V}^{n+1}-\vec{V}^{n}}{\Delta t}=-\frac{1}{\rho}\nabla_h p-\Pi_h^n, \tag{3.1}$$

$$\nabla_h\cdot\vec{V}^{n+1}=0, \tag{3.2}$$

where n refers to the time level, $\Delta t=t^{n+1}-t^n$ is the time step value and ∇_h denotes any numerical approximation for the divergence or the gradient operator. Π_h is the numerical approximation for all the remaining terms in equation (2.21) and are discussed in detail in the forthcoming sections.

In the first step of the projection method, known as the predictor step, an intermediate velocity field, denoted by $\vec{V}$, is obtained by marching explicitly in time. That is

$$\frac{\vec{V}^{*}-\vec{V}^{n}}{\Delta t}=\Pi_h^n, \tag{3.3}$$

with all the calculations performed with the *n*th time level values. The intermediate velocity field obtained in this step does not necessarily satisfy the continuity equation. In the second step, known as the projection step, the intermediate velocity field is corrected to yield a divergence-free velocity field by adding the pressure gradient:

$$\frac{\vec{V}^{n+1}-\vec{V}}{\Delta t}=-\frac{1}{\rho^n}\nabla_h p^{n+1}. \tag{3.4}$$

Applying ∇_h to both sides of the above equation yields

$$\nabla_h\cdot\left(\frac{1}{\rho^n}\nabla_h p^{n+1}\right)=\frac{1}{\Delta t}\nabla_h\cdot\vec{V}\ , \tag{3.5}$$

where equation (3.2) is used to enforce continuity and eliminate $\vec{V}^{n+1}$. Equation (3.5) is known as the Pressure Poisson Equation (PPE) and requires a system of equations to be solved for a solution to the pressure field. Once the pressure has been found, the advance time level velocity $\vec{V}^{n+1}$ can be determined via equation (3.4). It is important to note that ρ in equation (3.5) is kept inside the divergence operator since for multiphase flows density is not a constant. In fact, the variation of ρ in space results in a term proportional to density gradient, which contributes to the pressure within the free-surface transition region (D. B. Kothe et al., 1991).

3.2.2 Spatial Discretization

For the purpose of this section, it is convenient to expand equation (2.21) and re-write it for each space coordinate:

$$\frac{\partial u}{\partial t}+\frac{\partial(uu)}{\partial x}+\frac{\partial(vu)}{\partial y}=\frac{1}{\rho}\left(-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}(\mu\tau_{xx})+\frac{\partial}{\partial y}(\mu\tau_{xy})+F_{svx}\right)+g_x, \quad (3.6)$$

$$\frac{\partial v}{\partial t}+\frac{\partial(uv)}{\partial x}+\frac{\partial(vv)}{\partial y}=\frac{1}{\rho}\left(-\frac{\partial p}{\partial y}+\frac{\partial}{\partial y}(\mu\tau_{yy})+\frac{\partial}{\partial x}(\mu\tau_{yx})+F_{svy}\right)+g_y, \quad (3.7)$$

where τ_{xx}, τ_{yy}, τ_{xy} and τ_{yx} are the strain rate tensor terms defined as

$$\tau_{xx}=2\frac{\partial u}{\partial x}, \qquad \tau_{yy}=2\frac{\partial v}{\partial y}, \qquad \tau_{xy}=\tau_{yx}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}. \quad (3.8)$$

Likewise, the continuity equation can be written as

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0. \quad (3.9)$$

Here, F_{svx} and F_{svy} are the x and y components of the surface tension force.

Spatial discretization of the Navier-Stokes equations is completed through a finite-volume (FV) method for the convective terms and a finite-difference (FD) method for the viscous and surface tension terms. In FV method the governing equations are integrated over a control volume (CV) and are converted into surface integrals, or edge fluxes, employing the divergence theorem (Tryggvason & Balachandar, 2007). FV method can accommodate complex geometries and is conservative locally and globally

(Versteeg & Malalasekra, 2007). In FD method the differential form of the conservation equations are used to approximate the spatial derivatives in terms of their nodal values in each cell (Ferziger & Peric, 2002). FD method is easy to implement and can be used to obtain high-order schemes on regular structured grids (Ferziger & Peric, 2002).

Only the spatial discretization of equation (3.6) is discussed in the following sections. Discretization of equation (3.7) can be formulated analogously.

3.2.2.1 Computational Mesh

Spatial approximation of the governing equations starts by defining a mesh covering the entire computational domain over which the partial derivatives are approximated for each computational cell. Here for simplicity only a two-dimensional rectangular domain is considered with a computational mesh consisting of a fixed, two-dimensional, Eulerian, fully scattered, orthogonal grid with constant size rectangular cells as illustrated in Figure 3.1. Cell dimensions are Δx in the x direction and Δy in the y direction. Scalar quantities such as pressure and volume fraction are stored at the center of each computational cell and vector quantities such as velocity components are stored at cell edges. Indices i and j correspond to the column and row numbers, respectively, and are used to specify the location of the cell centered parameters. Likewise, $i+1/2$ and $j+1/2$ are used to identify the location of the cell edge parameters. This type of grid arrangement, known as staggered grid, makes it possible for several quantities to be readily approximated using appropriate difference methods without any need for interpolation (Ferziger & Peric, 2002; Versteeg & Malalasekra,

2007). The arrangement also provides a strong coupling between the velocity and the pressure which in turn helps to avoid some types of convergence problems and oscillation in pressure and velocity fields (Ferziger & Peric, 2002). The grid also includes a layer of fictitious cells around the entire mesh in which the boundary conditions are stored. Calculation of velocity components is defined over their respective control volumes, defined in Figure 3.1. Numerical approximation of scalar quantities such as pressure is performed over the CV defined by the cell boundaries.

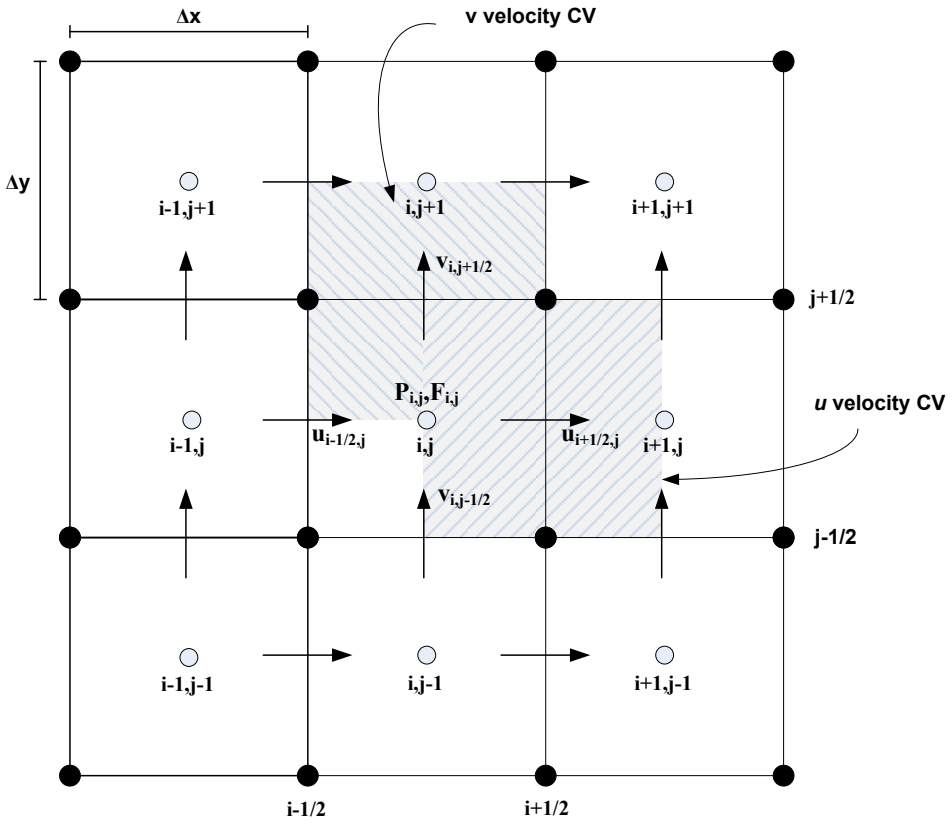

Figure 3.1: Staggered grid arrangement and the location of u and v -velocity CV's.

3.2.2.2 Convective Terms

The FV method employed here for the spatial discretization of the convective terms belongs to the total variation diminishing (TVD) class of schemes. A characteristic of TVD schemes that is particularly suitable for numerical simulation of multiphase flows is their ability to maintain high accuracy in smooth regions of the flow while being able to capture sharp non-oscillatory transitions at discontinuities (Yeoh & Tu, 2010). In other words TVD schemes are monotonicity preserving which implies that they do not create new undershoots and overshoots in the solution or accentuate exciting extremes (Versteeg & Malalasekra, 2007). The total variation of a grid function $\phi_{i+1/2,j}$ is defined by

$$TV(\phi) = \sum_j \sum_i \left| \phi_{i+3/2,j} - \phi_{i+1/2,j} \right|, \tag{3.10}$$

and the scheme is said to be TVD if

$$TV\left(\phi^{n+1}\right) \le TV\left(\phi^{n}\right), \tag{3.11}$$

where n refers to the time level (Zijlema & Wesseling, 1995). $\phi_{i+1/2,j}$ is considered to be monotone for all i's if

$$\min\left(\phi_{i-1/2,j}, \phi_{i+3/2,j}\right) \le \phi_{i+1/2,j} \le \max\left(\phi_{i-1/2,j}, \phi_{i+3/2,j}\right). \tag{3.12}$$

In reference to Figure 3.1, the general form of an upwind-biased discretization scheme written for a field parameter ϕ that is stored at the center of u-velocity CV and is to be

transported by the edge velocity $\langle u_{i,j} \rangle > 0$ can be written as the sum of a diffusive first-order upwind term and an anti-diffusive one as follows

$$\phi^{+}_{i,j} \approx \phi_{i-1/2,j} + \frac{1}{2}\psi\left(r^{+}_{i,j}\right)\left(\phi_{i+1/2,j} - \phi_{i-1/2,j}\right), \tag{3.13}$$

where

$$r^{+}_{i,j} = \left(\frac{\phi_{i-1/2,j} - \phi_{i-3/2,j}}{\phi_{i+1/2,j} - \phi_{i-1/2,j}}\right). \tag{3.14}$$

For $\langle u_{i,j} \rangle < 0$ the above equations can be written as

$$\phi^{-}_{i,j} \approx \phi_{i+1/2,j} + \frac{1}{2}\psi\left(r^{-}_{i,j}\right)\left(\phi_{i-1/2,j} - \phi_{i+1/2,j}\right), \tag{3.15}$$

where

$$r^{-}_{i,j} = \left(\frac{\phi_{i+3/2,j} - \phi_{i+1/2,j}}{\phi_{i+1/2,j} - \phi_{i-1/2,j}}\right). \tag{3.16}$$

Here, the nature of the scheme depends on the value of $\psi(r)$ which is known as the limiter function. For $\psi(r) = 0$ the first-order upwind differencing scheme is obtained while for $\psi(r) = 1$ the second-order central differencing scheme is recovered. For a scheme to be TVD, $\psi(r)$ must satisfy the TVD constraint defined by equation (3.11). In terms of $\psi - r$ relationship, If $0 < r < 1$ a TVD scheme satisfies $\psi(r) \le 2r$. If $r \ge 1$ the TVD constraint becomes $\psi(r) \le 2$. This can be better understood using the $\psi - r$

diagram shown in Figure 3.2 with the corresponding $\psi - r$ relationships for some of the most commonly used TVD limiter functions listed in Table 3-1. For a limiter function to be second-order accurate the scheme should pass through the point (1,1) in the $\psi - r$ diagram (Sweby, 1984) . A more complete discussion on TVD methods and other limiter functions can be found in Zijlema (1995) and Sweby (1984).

Table 3-1: Limiter functions and their definition.

Name	Limiter function definition $\psi(r)$
Van Leer	$\psi(r) = \frac{r + \lvert r \rvert}{1 + \lvert r \rvert}$
QUICK	$\psi(r) = \max\left[0, \min\left(2r, (3+r)/4, 2\right)\right]$
UMIST	$\psi(r) = \max\left[0, \min\left(2r, (1+3r)/4, (3+r)/4, 2\right)\right]$
SUPERBEE	$\psi(r) = \max\left[0, \min\left(2r, 1\right), \min\left(r, 2\right)\right]$

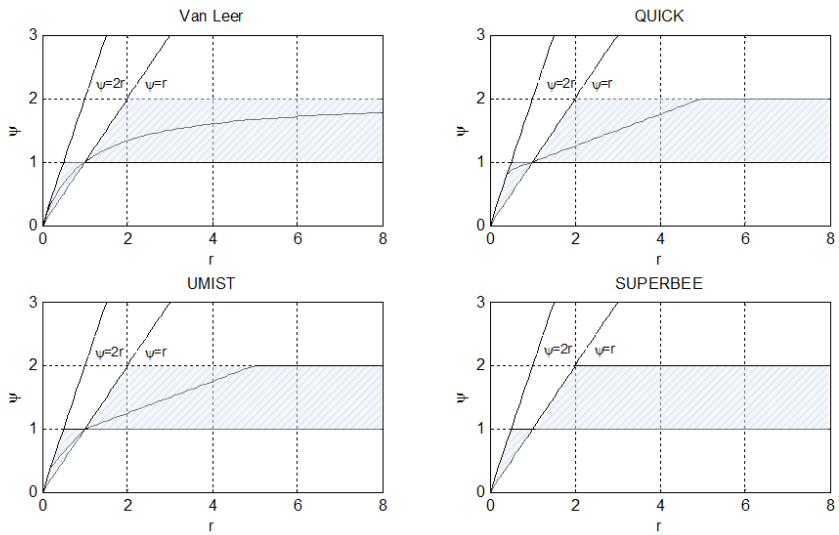

Figure 3.2: Limiter functions in $\psi - r$ diagram. Shaded area corresponds to second-order TVD region.

The transporting velocity or the so-called flux velocity, $\langle u_{i,j} \rangle$, corresponding to the quantity defined by equation (3.13) can calculated at the CV cell edge via a simple linear interpolation

$$\langle u_{i,j} \rangle \approx \frac{u_{i-1/2,j} + u_{i+1/2,j}}{2}. \tag{3.17}$$

Thus the first convective term in equation (3.6) written in its conservative form can be approximated using an upwind TVD method as follows

$$\left(\frac{\partial (uu)}{\partial x} \right)_{i+1/2,j} \approx \frac{u_{i,j} \langle u_{i,j} \rangle - u_{i-1,j} \langle u_{i-1,j} \rangle}{\Delta x}, \tag{3.18}$$

where $u_{i,j}$ is calculated by substituting ϕ with u in equation (3.13). The remaining terms located at the $i-1$ edge of the CV are calculated similarly. The second convective term can be approximated by considering interpolated v velocities on the top and on the bottom edges of the u-velocity CV as the flux velocities and calculating u using equation (3.13) acting along the y direction. Here, the conservative form of the governing equations is maintained as it produces more accurate results compared to the non-conservative form (D. B. Kothe et al., 1991). The TVD scheme described herein is second-order accurate away from discontinuities and reduces to first-order accurate near discontinuities where oscillations are to be avoided.

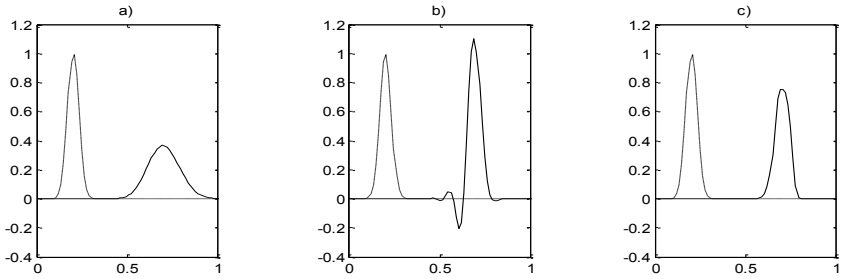

Figure 3.3: One-dimensional advection test $u_t + cu_x = 0$. Original wave profile (dashed line) is $u(x,0) = (\sin(\pi x))^{100}$, advection speed c = 0.5 and grid spacing is 350/5. Advected wave profile (solid line) after t = 1 for a) first-order upwind scheme, b) QUICK scheme with no limiter, c) QUICK scheme with TVD limiter, are shown

Figure 3.3 compares the performance various schemes for the one-dimensional advection of a wave profile. As expected, in the first-order upwind differencing scheme the solution is deteriorated over time due to false diffusion while high-order upwind schemes with no limiter produce oscillations near discontinuities. Clearly, the TVD scheme offers a compromise in terms of stability and accuracy.

3.2.2.3 Viscous Terms

The finite difference approximation of the derivatives involved in the calculation of the viscous terms is obtained using the standard central-differencing method. The strain rate tensor terms involved in equation (3.6), τ_{xx} and τ_{xy}, can be discretized for the u-velocity CV defined in Figure 3.1 as

$$\tau_{xx\ i+1,j} \approx \frac{2\left(u_{i+3/2,j} - u_{i+1/2,j}\right)}{\Delta x}, \tag{3.19}$$

on the right edge $(i+1, j)$, and as

$$\tau_{xy\ i+1/2,j+1/2} \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta x} + \frac{v_{i+1,j+1/2} - v_{i,j+1/2}}{\Delta y}, \tag{3.20}$$

on the top edge $(i+1/2, j+1/2)$ of the CV. Other edge values are calculated similarly.

Thus, the finite difference approximation of the viscous terms included in equation (3.6) can be presented as follows

$$\left(\frac{\partial}{\partial x}(\mu\tau_{xx}) + \frac{\partial}{\partial y}(\mu\tau_{xy})\right)_{i+1/2,j} \approx \frac{(\mu\tau_{xx})_{i+1,j} - (\mu\tau_{xx})_{i,j}}{\Delta x} + \frac{(\mu\tau_{xy})_{i+1/2,j+1/2} - (\mu\tau_{xy})_{i+1/2,j-1/2}}{\Delta y}, \tag{3.21}$$

where μ is to be interpolated to the appropriate CV edges and corners.

3.2.2.4 Surface Tension Term

Contrary to other momentum transport terms, surface tension forces are evaluated at the scalar CV centers and are subsequently interpolated to the appropriate CV edges via cell-centered value of the two neighbouring cells. This approach is found to yield more accurate results especially when dealing with coarser grids (D. B. Kothe et al., 1991). The definition of the CSF surface tension volume force was presented in section 2.5 and is repeated here with more details.

The surface tension force for cell i, j is given as

$$\vec{F}_{sv\ i,j} \approx \sigma k_{i,j} \vec{n}_{i,j}, \tag{3.22}$$

where, $\vec{n} = n_x\hat{i} + n_y\hat{j}$ is the inward normal vector pointing towards the $F = 1$ field and is defined as

$$\vec{n} \approx \nabla F. \tag{3.23}$$

The cell centred $\vec{n}$ can be expressed as the average of the corresponding vertex values (see Figure 3.4) calculated as follows for the top right corner of cell i, j

$$\begin{aligned}\vec{n}_{i+1/2,j+1/2} \approx & \frac{1}{2}\left(\left(F_{i+1,j+1} - F_{i,j+1}\right) + \left(F_{i+1,j} - F_{i,j}\right)\right)\hat{i} \\ & + \frac{1}{2}\left(\left(F_{i+1,j+1} - F_{i+1,j}\right) + \left(F_{i,j+1} - F_{i,j}\right)\right)\hat{j}.\end{aligned} \tag{3.24}$$

Other vertex values can be calculated similarly. Thus, $\vec{n}_{i,j}$ is given by

$$\vec{n}_{i,j} \approx \frac{1}{4}\left(\vec{n}_{i+1/2,j+1/2} + \vec{n}_{i+1/2,j-1/2} + \vec{n}_{i-1/2,j+1/2} + \vec{n}_{i-1/2,j-1/2}\right). \tag{3.25}$$

The local mean curvature k is also evaluated at the cell centers and can be expressed in terms of the unit normal vector $\hat{n} = \vec{n} / \|\vec{n}\|$ as per the following expression

$$k \approx \left(\nabla \cdot \hat{n}\right) = \frac{1}{\|\vec{n}\|}\left(\nabla \cdot \vec{n} - \frac{\left(\nabla \cdot \|\vec{n}\|\right)\left(\vec{n} \cdot \nabla\|\vec{n}\|\right)}{\|\vec{n}\|^2}\right), \tag{3.26}$$

where $\|\vec{n}\|$ is the magnitude of $\vec{n}$. To avoid finite differencing the square root associated with the calculation of $\|\vec{n}\|$, the spatial derivatives of $\|\vec{n}\|$ can be evaluated analytically (D. B. Kothe et al., 1991), leading to the following expression for k

$$k_{i,j} \approx \frac{1}{\|\vec{n}\|_{i,j}} \left(\left(1 - \frac{n_x^2}{\|\vec{n}\|} \right) \frac{\partial n_x}{\partial x} - \frac{n_x n_y}{\|\vec{n}\|} \left(\frac{\partial n_x}{\partial y} + \frac{\partial n_y}{\partial x} \right) - \left(1 - \frac{n_y^2}{\|\vec{n}\|} \right) \frac{\partial n_y}{\partial y} \right)_{i,j} . \quad (3.27)$$

The derivatives of $\vec{n}$ can easily be calculated from knowledge of $\vec{n}$ at cell vertices. For instance applying standard central-differencing yields

$$\left(\frac{\partial n_x}{\partial x} \right)_{i,j} \approx \frac{n_{x\,i+1/2,j+1/2} + n_{x\,i+1/2,j-1/2} - n_{x\,i-1/2,j+1/2} - n_{x\,i-1/2,j-1/2}}{2\Delta x_i}. \quad (3.28)$$

Therefore, for $\vec{F}_{sv} = F_{svx}\hat{i} + F_{svy}\hat{j}$ the face-centered values can be calculated by interpolating from the two nearest cell-centered values, giving

$$F_{svx\ i+1/2,j} \approx \frac{F_{svx\ i,j} + F_{svx\ i+1,j}}{2} , \quad (3.29)$$

for the x-momentum. The y-momentum value F_{svy} is calculated similarly.

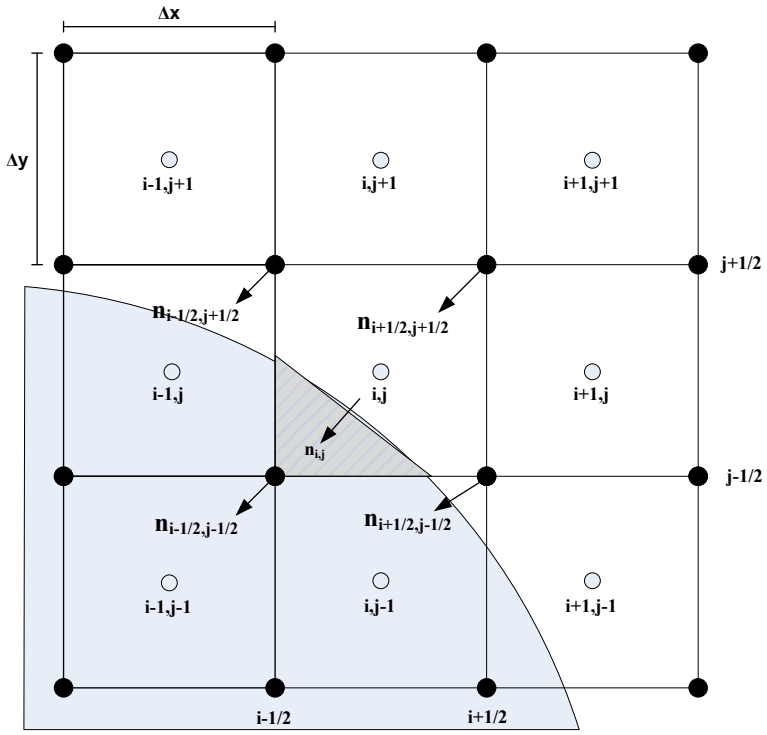

Figure 3.4: Approximation of the cell centered normal based on the cell vertex normal values.

A major difficulty associated with the presentation of the surface tension forces in the CSF model is the presence of the so-called spurious or parasite currents (Scardovelli & Zaleski, 1999). These currents are vortices appearing near the interface despite the absence of any external forces (Scardovelli & Zaleski, 1999). They arise from the abrupt transition of the F field across the interface leading to the inaccurate discrete approximations to the interface normal and curvature and do not disappear with mesh refinement (Renardy, Renardy, & Cristini, 2002).

One way to suppress spurious currents is to replace the volume fraction field used for the calculation of the surface tension forces with a smoothed volume fraction function which varies from 0 to 1 over a distance of the order of the mesh (Renardy et al., 2002).

The smoothing is performed in two dimensions by applying the following finite Laplacian filter (Lafaurie, Nardone, Scardovelli, Zaleski, & Zanetti, 1994):

$$\left[\mathcal{F}(F)\right]_{i,j} = \frac{1}{2}F_{i,j} + \frac{1}{8}\left(F_{i,j-1} + F_{i,j+1} + F_{i-1,j} + F_{i+1,j}\right). \tag{3.30}$$

The action of this filter may be repeated m_f times to obtain the desired degree of smoothing with the optimal number of filter application being $m_f = 1$ or 2 (Lafaurie et al., 1994). Thus, the smoothed volume fraction $\tilde{I}$ is defined as

$$\tilde{I} \quad \mathcal{F} \quad _{-}{}^{r}). \tag{3.31}$$

Smoothing the volume fraction effectively redistributes the surface tension forces over a thicker region near the interface and thus needs to be applied with care in order to prevent unphysical effects appearing in the simulation (D. B. Kothe et al., 1991).

3.2.2.5 Pressure Poisson Equation

The PPE equation obtained in the projection method (equation (3.5)) can be discretized in space using standard central-differencing. This gives,

$$\begin{aligned}\left(\nabla_h \cdot \left(\frac{1}{\rho^n}\nabla_h p^{n+1}\right)\right)_{i,j} \approx{}& \frac{1}{\Delta x^2}\left(\frac{1}{\rho^n_{x\,i+1/2}}\left(p^{n+1}_{i+1} - p^{n+1}_i\right) - \frac{1}{\rho^n_{x\,i-1/2}}\left(p^{n+1}_i - p^{n+1}_{i-1}\right)\right) \\ &+ \frac{1}{\Delta y^2}\left(\frac{1}{\rho^n_{y\,j+1/2}}\left(p^{n+1}_{j+1} - p^{n+1}_j\right) - \frac{1}{\rho^n_{y\,j-1/2}}\left(p^{n+1}_j - p^{n+1}_{j-1}\right)\right),\end{aligned} \tag{3.32}$$

for the left hand side, and

$$\left(\frac{1}{\Delta t}\nabla_h \cdot \vec{V}\right)_{i,j} \approx \frac{}{\Delta t}\left(\frac{u^*_{i+1/2,j} - u^*_{i-1/2,j}}{\Delta x} + \frac{v^*_{i,j+1/2} - v^*_{i,j-1/2}}{\Delta y}\right), \tag{3.33}$$

for the right hand side. Cell face values of density ρ_x and ρ_y are interpolated from the neighbouring cells in the x and y direction, respectively. For instance

$$\rho_{x\,i+1/2} = \frac{1}{2}(\rho_i + \rho_{i+1}). \tag{3.34}$$

Implicit treatment of pressure in equation (3.5) results in a linear system of equations with a pentadiagonal sparse variable coefficient matrix that needs to be solved to complete the projection step. The system of equations to be solved can be written as

$$Ap^{n+1} = b, \tag{3.35}$$

where the coefficient matrix A is obtained by applying equation (3.32) to all the cells and treating pressure p^{n+1} as the unknown, and the right hand side matrix b is given by equation (3.33) applied over the entire domain.

3.3 Numerical Implementation of the VOF Method

Once the velocity and pressure fields are updated one can proceed with updating the location of the interface according to the new flow field values. In the present work, this task is accomplished using the PLIC-VOF method discussed in section 2.4.1.5.

Briefly, in the VOF method each computational cell is associated with the volume fraction of dark fluid within that cell that is bounded by

$$0 \leq F_{i,j} \leq 1 \tag{3.36}$$

and is defined as

$$F_{i,j} = \begin{cases} 1, & \text{for cells filled with dark fluid} \\ > 0, < 1, & \text{for interface cells} \\ 0, & \text{for cells filled with light fluid.} \end{cases} \tag{3.37}$$

Two main steps were identified for completing the PLIC-VOF method: the interface reconstruction step where an approximation to the interface shape and location is inferred from the volume fraction field, and the VOF advection step wherein the interface is advanced in time using the VOF advection equation. These steps are elaborated in the following sections.

3.3.1 Interface Reconstruction Algorithm

The interface reconstruction method presented herein belongs to the PLIC family of reconstruction methods and is based on the algorithm outlined in Rudman (1997) which is an implementation of Youngs`s PLIC method. Except for the calculation of the interface normals, the PLIC algorithm is entirely local to each individual cell and as such references to cell indices are omitted in this subsection for simplicity.

As stated previously, in the PLIC method the interface is reconstructed as a series of straight line segments within each computational cell. Each line segment is defined by a slope and two intercepts with the local cell edges. The slope of the line is given by the interface normal and the intercepts follow from invoking volume conservation (Rider & Kothe, 1998). The algorithm used to determine the interface normal within each cell for

the PLIC method is identical to the method described in section 3.2.2.4 for the calculation of the interface curvature. However, unlike the curvature calculation, the un-soothed volume fraction function F is used at all times for the interface reconstruction to maintain the sharp interface presentation.

Following the calculation of the interface normal the interface is rotated such that the portion of the cell containing dark fluid is positioned on the bottom right of each cell. This is equivalent to flipping the interface in such way that n_x becomes positive and n_y becomes negative (Figure 3.5). It is important to keep track of the rotation for each cell to assign the correct outward fluxes in the advection step. Rotating the interface allows for the number of possible interface configurations within a cell to be reduced to four cases as depicted in Figure 3.6.

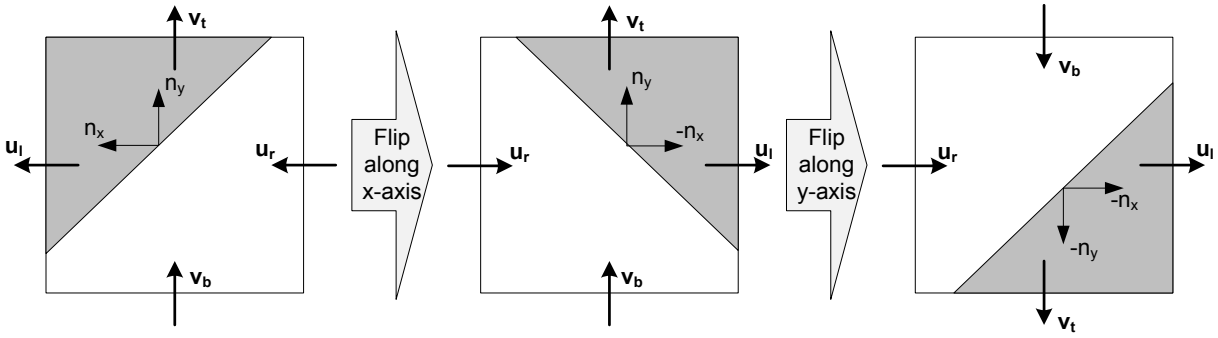

Figure 3.5: Rotating the interface within the cell. Notice that the location of the edge velocities change as the interface rotates.

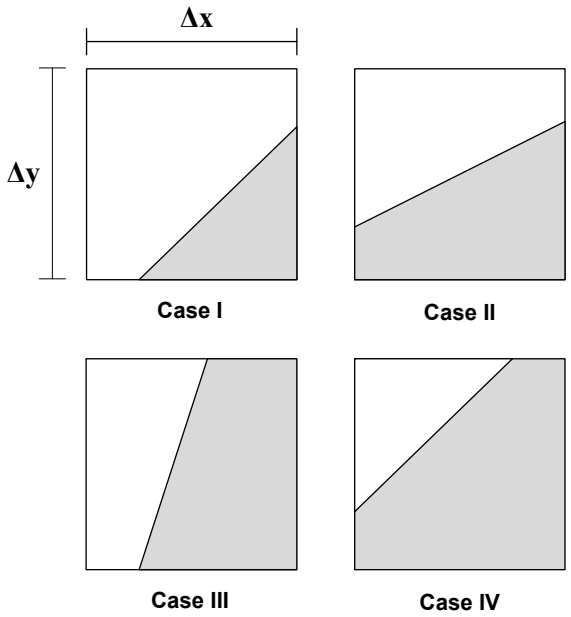

Figure 3.6: The four possible interface reconstruction cases after interface rotation.

The line cutting the cell can be presented by the equation:

$$n_x x + n_y y = d \tag{3.38}$$

where *d* is a constant. The angle the interface makes with the positive x-axis is

$$\beta = \tan^{-1}\left(\frac{|n_x|}{|n_y|}\right), \tag{3.39}$$

where the absolute values of the normal components are used. Here, $0 \le \beta \le \pi/2$ due to the reorientation of the interface done previously. Conservation of volume requires the interface within the cell to be positioned such that the dark area under the interface becomes

$$A = F\Delta x\Delta y. \tag{3.40}$$

This is the constraint that needs to be enforced in order to determine the reconstruction case for an interface with a given β.

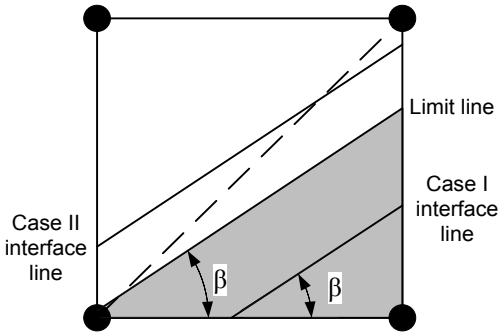

Figure 3.7: Reconstruction case selection for case I and II with $\alpha < \pi/4$.

Consider the case depicted in Figure 3.7, where for a given value of β the solid line cutting the cell from the bottom left corner is the limiting case separating case I and II. The area of the dark fluid under the limit line is

$$A_{\lim} = \frac{1}{2}\Delta x^2 \tan\beta. \tag{3.41}$$

Normalizing by the total area of the cell one obtains the corresponding volume fraction,

$$F_{\lim} = \frac{1}{2}\frac{\Delta x}{\Delta y}\tan\beta. \tag{3.42}$$

Defining the angle α to be

$$\alpha = \tan^{-1}\left(\frac{\Delta x}{\Delta y}\tan\beta\right), \tag{3.43}$$

equation (3.42) can be written as

$$F_{\lim} = \frac{1}{2}\tan\alpha. \tag{3.44}$$

It can be shown that the dashed line cutting the cell diagonally in half corresponds to $\alpha_{half} = \pi/4$. Thus for the case under consideration with $\alpha < \alpha_{half}$, if $F \le F_{\lim}$ the interface must lie below the limit line which corresponds to the reconstruction case I. On the other hand, if $F \le 1 - F_{\lim}$ the interface lies above the limit line invoking the reconstruction case II. Other cases can be analysed analogously. The logic used to determine each case is summarized in Table 3-2.

Table 3-2: Reconstruction case selection based on the α - F values.

	$\alpha < \pi/4$	$\alpha \geq \pi/4$
Case I	$F \leq \frac{1}{2}\tan\alpha$	$F \leq \frac{1}{2}\cot\alpha$
Case II	$F \leq 1-\frac{1}{2}\tan\alpha$	-
Case III	-	$F \leq 1-\frac{1}{2}\cot\alpha$
Case IV	otherwise	otherwise

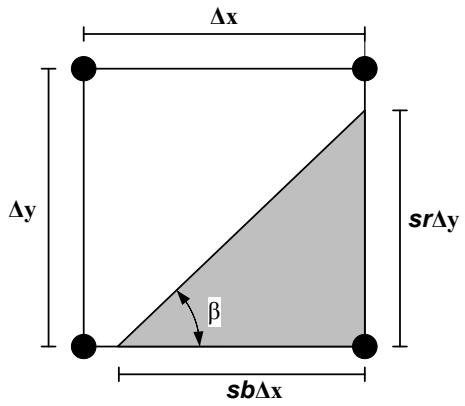

Figure 3.8: Reconstructed interface and its relevant parameters.

Once the case has been determined the reconstruction of the interface can be completed by calculating the side fractions. These are the fraction of the top, right, bottom and left edges of the cell that lie within the fluid and are denoted by st, sr, sb and sl, respectively. In the case depicted in Figure 3.8, sr and sb are related to β by the following equation:

$$\tan\beta = \frac{sr\Delta y}{sb\Delta x}. \tag{3.45}$$

Recalling equation (3.43), this yields

$$\tan\alpha = \frac{sr}{sb}. \tag{3.46}$$

Once again, conservation of volume gives

$$srsb = 2F. \tag{3.47}$$

Thus, solving for sb in equation (3.46) and inserting the results into equation (3.47) one obtains

$$sr = \left(2F\tan\alpha\right)^{1/2}. \tag{3.48}$$

Similarly sb can be calculated as

$$sb = \left(\frac{2F}{\tan\alpha}\right)^{1/2}. \tag{3.49}$$

Calculation of side fractions for other cases follows analogously. The intercepts can be calculated by multiplying each side fraction with its corresponding cell edge dimension. Table 3-3 summarizes the definition of the volume fraction for each reconstruction case.

Table 3-3: Side fraction calculation for each reconstruction case.

	Case I	Case II	Case III	Case IV
st	0	0	$F-\frac{1}{2}\cot\alpha$	$1-\left(2(1-F)\cot\alpha\right)^{1/2}$
sr	$\left(2F\tan\alpha\right)^{1/2}$	$F+\frac{1}{2}\tan\alpha$	1	1
sb	$\left(2F\cot\alpha\right)^{1/2}$	1	$F+\frac{1}{2}\cot\alpha$	1
sl	0	$F-\frac{1}{2}\tan\alpha$	0	$1-\left(2(1-F)\tan\alpha\right)^{1/2}$

3.3.2 The VOF Advection Algorithm

The VOF advection is governed by equation (2.23), which for a divergence free velocity field can be written as

$$\frac{\partial F}{\partial t} + \nabla \cdot \left(\vec{V} F \right) = 0. \tag{3.50}$$

This can be discretized in time and space using FV principles as

$$F_{i,j}^{n+1} = F_{i,j}^{n} + \frac{\Delta t}{\Delta x}\left(f_{i-1/2,j} - f_{i+1/2,j} \right) + \frac{\Delta t}{\Delta y}\left(f_{i,j-1/2} - f_{i,j+1/2} \right), \tag{3.51}$$

where, for instance, $f_{i-1/2,j} = \left(Fu \right)_{i-1/2,j}$ denotes the flux of F across the left edge of the cell $\left(i, j \right)$.

The simplest form of the advection algorithm is obtained when equations (3.51) is advanced in time using a fractional step or operate split method (Puckett et al., 1997). This results in the following form of the advection equation:

$$F_{i,j}^{*} = F^{n} + \frac{\Delta t}{\Delta x}\left(f_{i-1/2,j} - f_{i+1/2,j} \right), \tag{3.52}$$

$$F_{i,j}^{n+1} = F_{i,j}^{*} + \frac{\Delta t}{\Delta y}\left(f_{i,j-1/2}^{*} - f_{i,j+1/2}^{*} \right), \tag{3.53}$$

where F^{*} is the intermediate volume fraction field obtained after sweeping in the x-direction and f^{*} denotes the edge fluxes calculated based on the F^{*} field values. The general idea here is to reconstruct the interface using the F^{n} field data, calculate the

fluxes across cell edges and advect the F^n field in the x-direction using equation (3.52) to obtain the intermediate volume fraction field F^*. In the next step, the F^* field data is used to reconstruct the interface and calculate the fluxes while equation (3.53) is used to advect the F^* field in the y-direction and obtain the advanced time level volume fraction field F^{n+1}. Defining the advection equation in such a manner allows for the PLIC reconstruction algorithm and the advection algorithm to be implemented only in one dimension which greatly simplifies the task in hand from a programming perspective.

Equations (3.52) and (3.53) are exact for a divergence free velocity field. In practice, however, the velocity field used to calculate the fluxes has a none zero divergence due to the divergence criterion of the PPE solution (D. B. Kothe et al., 1991). Therefore, to maintain conservation of F a divergence correction term is added to the advection algorithm. This yields

$$F_{i,j}^{*}=\left(F^{n}+\frac{\Delta t}{\Delta x}\left(f_{i-1/2,j}-f_{i+1/2,j}\right)\right)\Bigg/\left(1-\frac{\Delta t}{\Delta x}\left(u_{i+1/2,j}-u_{i-1/2,j}\right)\right), \qquad (3.54)$$

for an implicit differencing of the x-sweep, and

$$F_{i,j}^{n+1}=F_{i,j}^{*}+\frac{\Delta t}{\Delta y}\left(v_{i,j+1/2}-v_{i,j-1/2}\right)+\frac{\Delta t}{\Delta y}\left(f_{i,j-1/2}^{*}-f_{i,j+1/2}^{*}\right), \qquad (3.55)$$

for an explicit differencing of the y-sweep (Puckett et al., 1997). Second-order time accuracy can be obtained for this procedure simply by alternating the sweep direction at each time step, taking care that the first sweep is differenced implicitly while the second one is differenced explicitly (Puckett et al., 1997).

The fluxes appearing in equations (3.54) and (3.55) can be calculated geometrically using the reconstructed interface data and the size of the time step. For instance, consider the reconstructed interface depicted in Figure 3.9, subject to a positive outward velocity on its right edge. It is trivial to show that the amount of dark fluid that crosses the right cell boundary during time $t = \Delta t$ is equal to the amount of the dark fluid contained in the rectangle confined by $u_r \Delta t$ and Δy. Thus, denoting this area by A_r, the fraction of the dark fluid advected across the right cell boundary is given by

$$f_r = u_r \left(\frac{A_r}{u_r \Delta t \Delta y} \right) = \frac{A_r}{\Delta t \Delta y}. \tag{3.56}$$

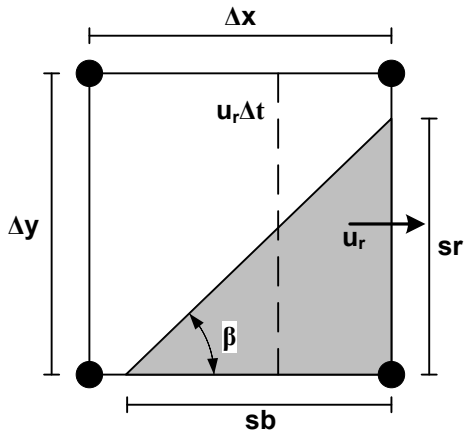

Figure 3.9: The fraction of the dark fluid to the right of the dotted line is advected into the neighbouring cell on the right.

The calculation of area A_r is carried on geometrically using the values of the side fractions and the slope angle β. For instance, for the case under consideration it can be shown that

$$A_r = u_r \Delta t \times sr \Delta y - 0.5\left(\left(u_r \Delta t\right)^2 \tan\beta\right). \tag{3.57}$$

Obviously, if $u_r \Delta t > sb\Delta x$ then $A_r = F\Delta x\Delta y$. Table 3-4 contains the calculation algorithm used to determine the areas involved in the x-sweep for all the interface reconstruction cases. Areas corresponding to the y-sweep can be calculated by interchanging the roll of x and u with y and v, respectively. In such a case, β measures the angle the interface makes with the positive y-axis. Due to the operator split advection algorithm, only the fluxes in the direction of the sweep need to be calculated, i.e. top and bottom fluxes are not considered for the x-sweep. Also, it is important to point out that the above mentioned procedure for calculating the fluxes is only applicable to the outward-pointing velocities, i.e. positive velocities for the right edge and negative velocities for the left edge of a given cell. This implies that the amount of flux through a cell boundary is determined using the volume fractions immediately upstream of the edge velocity.

Finally, numerical overshoot and undershoot may lead to some volume fractions that violate the range limits defined by equation (3.36). This can be rectified by truncating the values of volume fractions at the end of each sweep such that

$$F_{i,j} = \min\left(1, \max\left(F_{i,j}, 0\right)\right). \tag{3.58}$$

These adjustments render the F field slightly non-conservative but are typically a small percentage of the total volume and were found to have little effect on the final results.

Table 3-4: Flux area calculation. Outward-pointing velocities are defined as being positive and flux area calculation is not done for inward-pointing velocities.

	$u_l > 0$	$u_r > 0$
Case I	if $u_l \Delta t \le (1 - sb)\Delta x$ $A_l = 0$ else $A_l = 0.5 \tan\beta \left(u_l \Delta t - (1 - sb)\Delta x\right)^2$	if $u_r \Delta t \ge sb \Delta x$ $A_r = F \Delta x \Delta y$ else $A_r = u_r \Delta t \left(sr \Delta y - 0.5 u_r \Delta t \tan\beta\right)$
Case II	$A_l = u_l \Delta t \left(sl \Delta y - 0.5 u_l \Delta t \tan\beta\right)$	$A_r = u_r \Delta t \left(sr \Delta x - 0.5 u_r \Delta t \tan\beta\right)$
Case III	if $u_l \Delta t \le (1 - sb)\Delta x$ $A_l = 0$ else if $u_l \Delta t \le (1 - st)\Delta x$ $A_l = 0.5 \tan\beta \left(u_l \Delta t - (1 - sb)\Delta x\right)^2$ else $A_l = u_l \Delta t \Delta y - (1 - F)\Delta x \Delta y$	if $u_r \Delta t \le st \Delta x$ $A_r = u_r \Delta t \Delta y$ else if $u_r \Delta t \le sb \Delta x$ $A_r = u_r \Delta t \Delta y - 0.5 \tan\beta \left(u_r \Delta t - st \Delta x\right)^2$ else $A_r = F \Delta x \Delta y$
Case IV	if $u_l \Delta t \ge (1 - st)\Delta x$ $A_l = u_l \Delta t \Delta y - (1 - F)\Delta x \Delta y$ else $A_l = u_l \Delta t \left(sl \Delta y + 0.5 u_l \Delta t \tan\beta\right)$	if $u_r \Delta t \le st \Delta x$ $A_r = u_r \Delta t \Delta y$ else $A_r = u_r \Delta t \Delta y - 0.5 \tan\beta \left(u_r \Delta t - st \Delta y\right)^2$

3.4 Boundary Condition Implementation

As previously stated, boundary conditions are enforced using a layer of fictitious cells around the entire mesh in which the boundary conditions are stored. The values for dependent variables in the fictitious cells are set such that the desired boundary conditions are met at the boundaries. Boundary conditions for velocity, pressure and volume fraction are discussed herein.

3.4.1 Velocity Boundary Conditions

The appropriate velocity boundary conditions for solid walls is the no-slip condition ($u = v = 0$) (Versteeg & Malalasekra, 2007). For staggered grids, the location of the normal velocities coincides with the domain boundaries. As a result, normal velocities that coincide with solid walls are simply set to zero. On the other hand, prescription of tangential wall velocities cannot be accomplished explicitly since the location of tangential velocities is half grid space away from the boundary. Instead, fictitious cell velocities are set in such a way that linear interpolation would yield the desired wall tangential velocity boundary condition. Thus, for the example illustrated in Figure 3.10, the south wall velocity boundary condition is determined by

$$v_{i,1+1/2} = 0, \tag{3.59}$$

for the normal velocities, and

$$u_{i+1/2,3/2} = u_s = \frac{u_{i+1/2,2} + u_{i+1/2,1}}{2}, \tag{3.60}$$

for the tangential velocities, which for $u_s = 0$ gives

$$u_{i+1/2,1} = -u_{i+1/2,2}. \tag{3.61}$$

If the fluid immediately adjacent to the walls is assumed to travel at the same tangential velocity as the rest of the fluid the so-called free-slip condition is obtained. For the case depicted in Figure 3.10 this gives

$$u_s = u_{i+1/2,1} = u_{i+1/2,2}. \tag{3.62}$$

Velocity boundary condition at other walls can be implemented similarly.

In staggered grids, inflow velocity boundary condition can also be easily implemented by setting the tangential velocities at the inflow boundary to zero (similar to equation (3.61)) and assigning the desired inflow value to the corresponding normal velocities that reside on the inflow boundary.

Outflow velocity boundary condition, however, is more complicated to implement. Typically, it should be located far away from geometric disturbances where the flow is considered fully developed with a known flow direction (Versteeg & Malalasekra, 2007). Thus, a successful implementation of an outflow velocity boundary condition should disturb the upstream flow as little as possible and exit the domain gently. (Tryggvason & Balachandar, 2007). One way to accomplish this is by setting the normal gradient of the velocity at the outlet to zero. That is,

$$\left(\frac{\partial \vec{V}}{\partial n}\right)_{outlet} = 0, \tag{3.63}$$

where n is the direction normal to the outlet, i.e. x for an outflow boundary condition located at the east side of the grid.

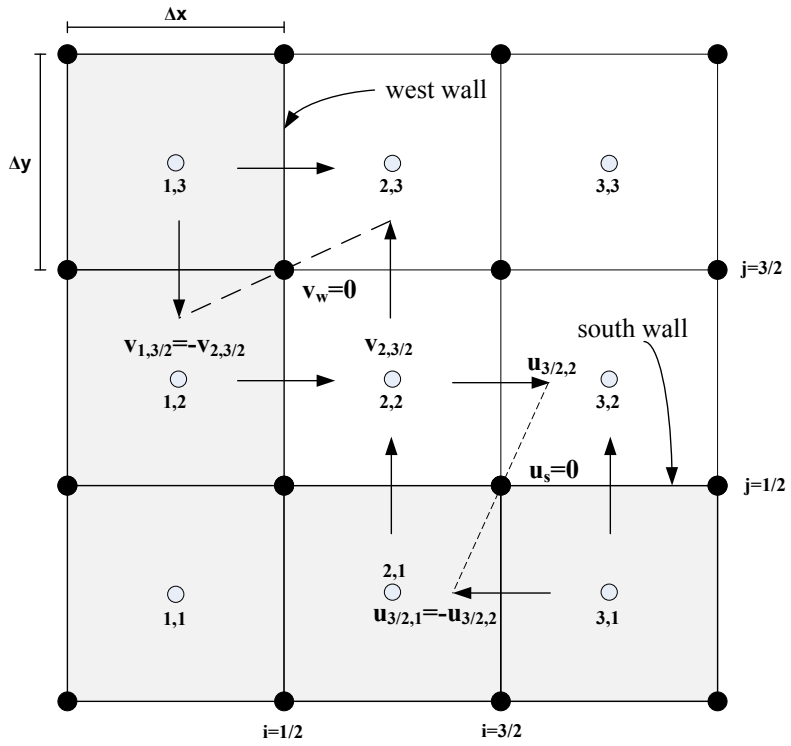

Figure 3.10: Example of velocity boundary condition implementation at the bottom left hand corner of a grid surrounded by solid walls. No-slip velocity boundary condition is enforced on the south wall while free-slip velocity boundary condition is enforced on the west wall. Normal velocities in both cases are explicitly set to zero while tangential velocities are set in such way that the desired tangential wall boundary values are obtained.

3.4.2 Pressure Boundary Conditions

When the mass flux through a boundary is prescribed, as is the case in this work, the mass flux correction in the PPE (equation (3.5)) is also zero there (Ferziger & Peric, 2002). This corresponds to a zero Neumann boundary condition for pressure at all mesh boundaries with a specified velocity boundary condition. Thus, denoting the mesh boundaries by Γ, the pressure boundary condition used in the PPE is given by

$$\left(\frac{\partial p}{\partial n}\right)_{\Gamma}^{n+1} = 0. \tag{3.64}$$

Finally, it can be shown that the velocity boundary conditions used for the intermediate velocity field calculation are equal to the time advance velocity field boundary conditions (Ferziger & Peric, 2002). That is,

$$\vec{V}_{\Gamma} = \vec{V}_{\Gamma}^{n+1}. \tag{3.65}$$

3.4.3 Volume Fraction Boundary Conditions

Calculation of the interface normal for the surface tension approximation and the interface reconstruction step requires the volume fraction field to be extrapolated to the fictitious cells to allow for the interface normals to be found at cells bordering the mesh boundaries. Thus, for instance, the boundary conditions for F at the west side of the mesh is given by

$$F_w = F_{1,j}. \tag{3.66}$$

Other mesh boundaries are treated similarly. Also, all other cell centered quantities such as ρ and μ are extended to the fictitious cells in a similar manner if needed.

3.5 Numerical Stability

A numerical solution is said to be stable if it does not magnify the errors that appear in the course of the numerical solution (Ferziger & Peric, 2002). Generally speaking, one often desires to employ the largest possible mesh and time increments in order to maximize the computational efficiency and time for a given computational domain. On the other hand, the mesh and time increments must be small enough to resolve expected spatial variations in all dependent variables and to ensure numerical stability,

respectively (Nichols et al., 1980). Thus, for a given mesh size the time step is to be selected such that numerical stability is guaranteed throughout the calculation process.

The algorithms described above for the approximation of the Navier-Stokes equation is explicit in nature and as such is subject to its respective time step constraints. Linear stability analysis as performed in Ferziger (2002) for the discretization of the convective and diffusive terms results in the following time step constraint:

$$\Delta t_{CD} = \left(\frac{4\nu_{\max}}{\Delta x^2} + \frac{4\nu_{\max}}{\Delta y^2} + \frac{|u|_{\max}}{\Delta x} + \frac{|v|_{\max}}{\Delta y} \right)^{-1}, \tag{3.67}$$

where $\nu = \mu / \rho$ is the coefficient of the kinematic viscosity. Time constraint analysis associated with the CSF model is derived in Brackbill (1992) and is formulated as

$$\Delta t_{CSF} = \left(\frac{0.5(\rho_1 + \rho_2)\Delta x^3}{2\pi\sigma} \right)^{1/2}. \tag{3.68}$$

Also, fluid particles cannot move more than one cell in a single time step (Ferziger & Peric, 2002) which results in the so-called CFL condition defined as

$$\Delta t_{CFL} = C_{CFL} \min\left(\frac{\Delta x}{|u|_{\max}}, \frac{\Delta y}{|v|_{\max}} \right), \tag{3.69}$$

where C_{CFL} is known as the Courant number and should be less than 1 in theory. The CFL condition also applies to the VOF advection equation requiring C_{CFL} to be less than 0.5 in order to avoid calculating overlapping fluxes out of a cell.

Therefore, to ensure the numerical stability of the numerical scheme described in the previous sections, the maximum time step size used to advance the governing equations in time for a given computational cycle must be selected such that

$$\Delta t < \min\left(\Delta t_{CD}, \Delta t_{CSF}, \Delta t_{CFL}\right). \tag{3.70}$$

3.6 Programming Environment and the Numerical Method Implementation

Although the selection of the programming environment, to a large extend, is a matter of personal preferences of the implementer, the following discussion is presented to specify some of the observations made by the author that influenced the selection of the programming environment.

Very often the numerical study of a physical phenomenon is overshadowed by the difficulties associated with implementation of the selected numerical scheme. Although historically programming languages such as C, C++ and FORTRAN have been used to develop scientific codes, a big portion of these codes are dedicated to the development of tools and functions that would allow for the visualization of the generated data in the form of charts, graphs, velocity or pressure fields, etc. Therefore, much effort needs to be put into programming tools that have no significance for the actual study and in many cases require programming skills that surpass those needed for the proper implementation of the numerical scheme only. Codes created in such environments also need to be manipulated to incorporated new features whenever additional data analysis features are required. In such cases the programming skills of the developers largely

determine the proper presentation of the results and hence to a certain degree the study itself.

In addition, in recent years, progress in parallel computing has allowed for complex physical phenomena to be simulated efficiently using more accurate and sophisticated numerical schemes. However, coding in parallel computing environments often require special knowledge of the implementation platform and ultimately involves significant code manipulation.

It is, therefore, advantageous for the numerical study to be performed in a programming environment that could provide built-in libraries for data processing, data visualization, mathematics, etc, and can be integrated in a parallel computing environment with no or little code manipulation. This could eventually help minimize the difficulties associated with raw programming and would allow for the numerical scheme and the physical phenomenon to become the prime focus of the study.

MATLAB is one such programming environment that has become the platform of choice for many academic and industrial studies in the recent years. It provides built-in functions that can easily manipulate and solve the large matrices and system of equations that often appear in engineering applications and continues to benefit from the growing number data analysis tools provided to it. It also supports parallel computing which allows users to take advantage of multiprocessor environments with little or no code changes (*Built-in parallel computing support in MathWorks products.*).

For instance, the discrete field quantities defined for the two-dimensional rectangular computational domain of this work can easily be presented in MATLAB with a series of matrices (Table 3-5). Boundary conditions can also be implemented by adding the appropriate rows and columns to the corresponding matrices. This becomes a trivial task in MATLAB since all the data is treated as matrices with many commands available to manipulate them. Finally, the solution to equation (3.35) obtained from the application of the PPE can easily be solved in MATLAB using many available solvers for linear systems. For example, the backslash operator (\) can efficiently accomplish this task by taking advantage of all the available processors. This is a significant advantage since finding the solution to equation (3.35) represents the most computationally expensive step of the simulation.

Table 3-5: Data structure for a $m_x \times m_y$ mesh resolution.

Variable name	Interior resolution	Resolution with boundary conditions
p and F	$m_x \times m_y$	$(m_x+2)\times(m_y+2)$
u -velocity	$(m_x-1)\times m_y$	$(m_x+1)\times(m_y+2)$
v -velocity	$m_x \times (m_y-1)$	$(m_x+2)\times(m_y+1)$

Each major part of the numerical scheme is coded in a separate function allowing it to be treated as an individual program module that can be modified without affecting the other parts of the program. The calling order of each module is determined by a main function that defines and initializes the simulation case and advances the solution in

time. Figure 3.11 illustrates the computational flow of the numerical scheme described in this section for one computational cycle.

The final code was written and run in MATLAB R2009.a on the HPC cluster at Concordia University which is built using the HP XC framework, running the 64-bit Linux operating system. Unfortunately, due to the limited access to the cluster environment and the automatic handling of parallel jobs by MATLAB a precise assessment of the performance of the code could not be made. However, it can be confirmed that the fully vectorized code was able to take advantage of a maximum of 7 processors for the falling liquid film simulation which resulted in a significant reduction in computational time compared to a single-processor run (at least by a factor of 6).

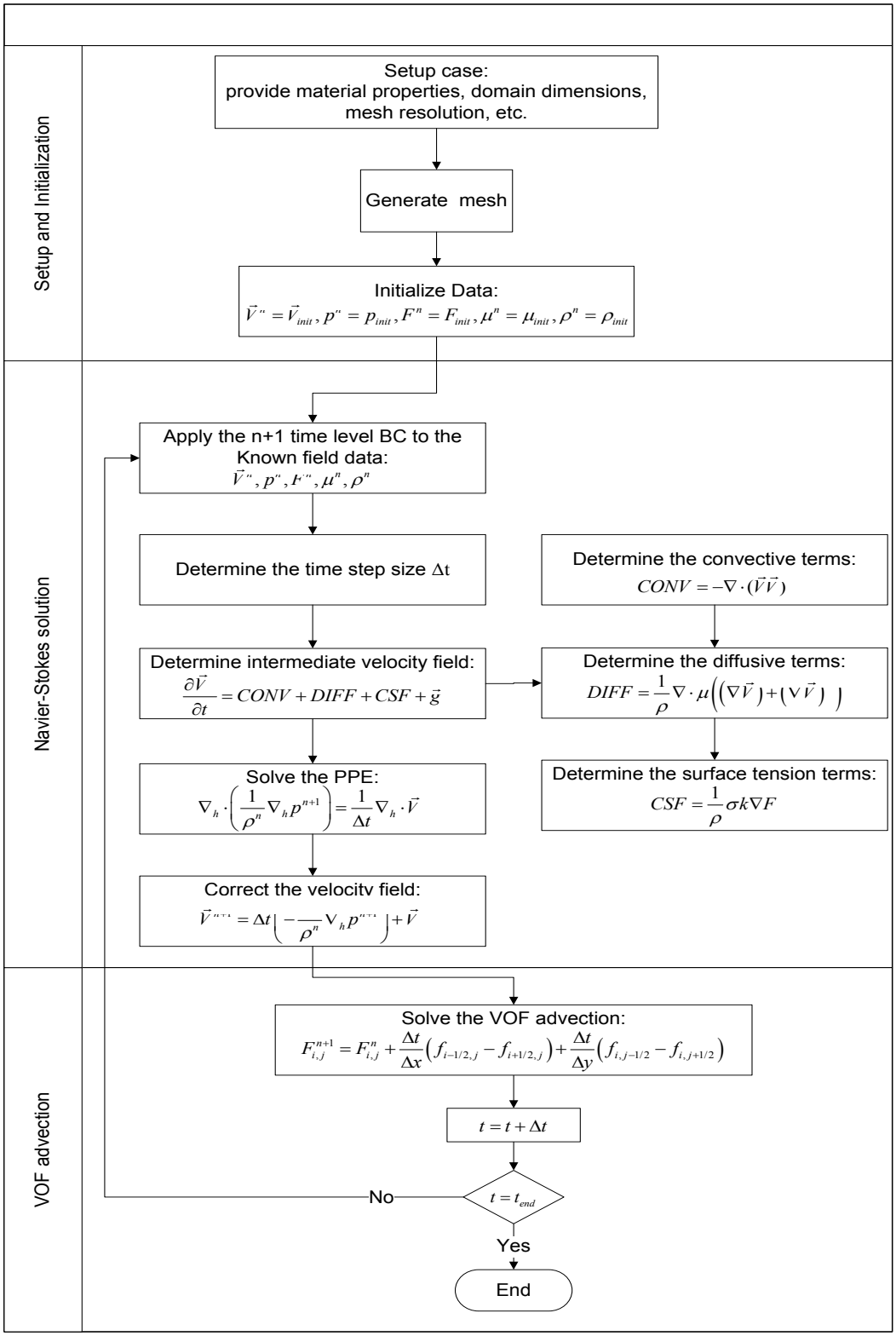

Figure 3.11: Numerical scheme implementation flowchart.

3.7 Summary

A detailed description of the numerical methods used to approximate the governing equations was presented. The multiphase Navier-Stokes equation was integrated in time using an explicit forward Euler method while a two step projection method was used to enforce continuity and advance the velocity field in time. TVD schemes were shown to yield better results for the spatial discretization of the convective terms compared to simple upwind or high-resolution schemes with no limiters. The diffusive terms were discretized using standard second-order central differencing and the surface tension term was approximated by calculating the interface normals at cell vertices. Cell centered interface curvature and surface tension forces were found using the averaged value of these cell vertex normals. The VOF advection equation was discretized using an operator split scheme with the volume fraction fluxes calculated using the geometrically reconstructed interface within each computational cell. Subsequently, implementation of the boundary conditions was discussed for solid wall, inflow and outflow boundaries. Numerical stability of the scheme was discussed and the time step constraints were presented. Finally, justifications were put forward for selecting MATLAB as the implementation environment and the post processing tool.

Chapter 4

Code Validation

4.1 Overview

In this chapter two benchmark tests are performed to validate the implementation of the PLIC-VOF method and the CSF model. This is followed by the simulation of Rayleigh-Taylor instability and a quantitative assessment of the overall accuracy of the algorithm through a grid refinement test.

4.2 Benchmark Tests

4.2.1 Standard VOF Advection Test

Following the discussion in Rudman (1997), an advection method is best evaluated in a shearing velocity field where stretching, shearing, fluid merging and breakup are possible. The following velocity field is selected:

$$u(x,y) = \cos(x)\sin(y), \qquad v(x,y) = -\sin(x)\cos(y), \tag{4.1}$$

with $x, y \in [0, \pi]$. The mesh size is 100 x 100. A circle with a radius of $\pi / 5$ is placed at the bottom of the enclosed domain as shown in Figure 4.1.

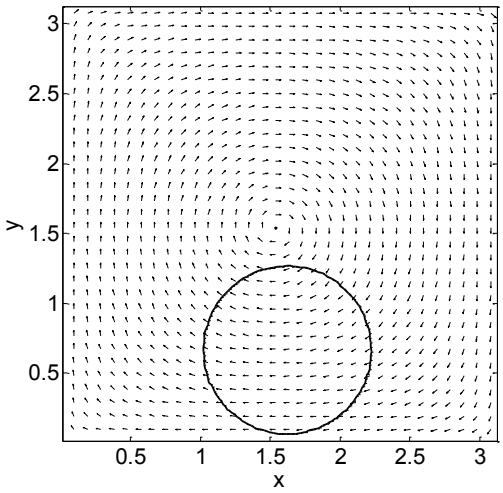

Figure 4.1: Initial F field configuration with the shearing flow velocity vectors. Velocity field is not shown at the grid resolution for clarity.

Selecting a Courant number of 0.25, the simulation is integrated forward in time for N steps before reversing the sign of the velocity field and integrating for an additional N steps. At the end of this procedure, a perfect advection scheme would return the initial F field. The simulation results after N=1000 and N=2000 steps are shown in Figure 4.2. Also, errors associated with different number of backward steps are presented in Table 4-1 for N=250, 500, 1000 and 2000. The error is evaluated as per the following equation:

$$E = \frac{\sum_{i,j=1}^{N_{mesh}} \left| F_{exact} - F_{estimated} \right|}{\sum_{i,j=1}^{N_{mesh}} F_{exact}}, \tag{4.2}$$

where the sum is over the entire computational mesh, F_{exact} is the initial F field, $F_{estimated}$ is the simulated field and N_{mesh} is the number of mesh cells (Rudman, 1997).

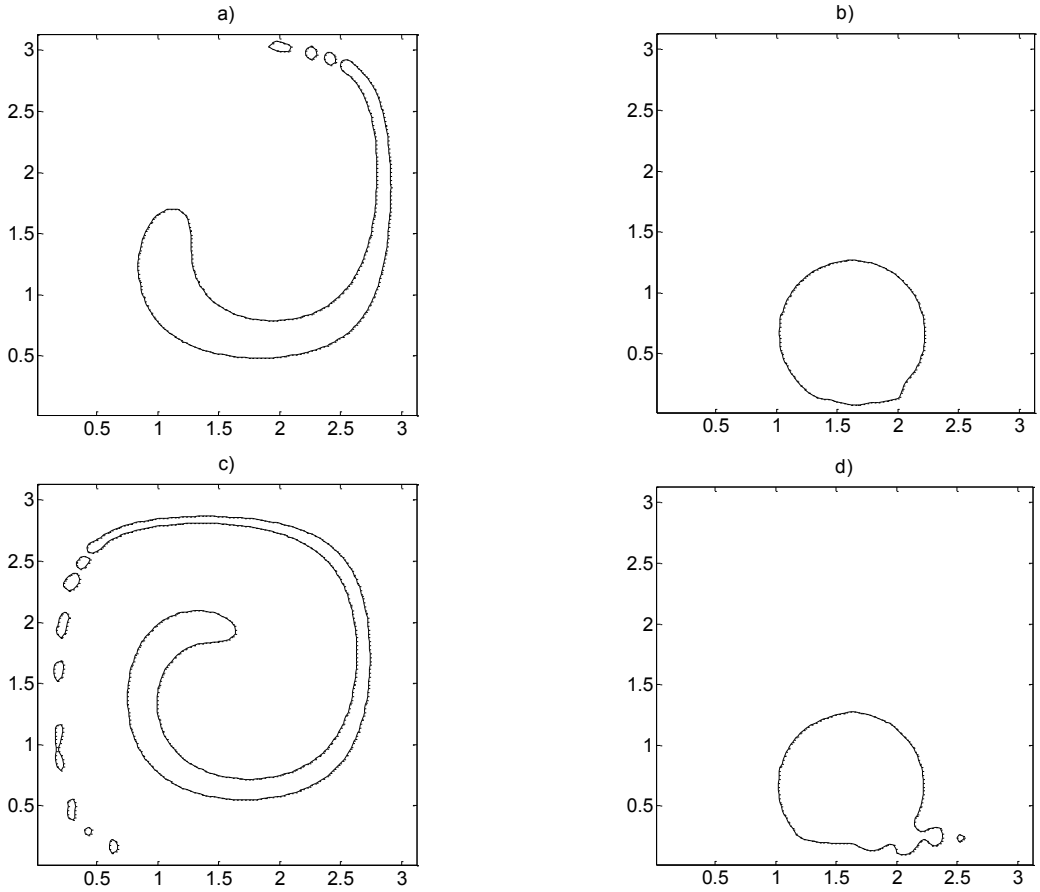

Figure 4.2: Contours of F = 1/2 for the results of the shearing field: a) after 1000 steps forward, b) after 1000 steps forward followed by 1000 steps backward, c) after 2000 steps forward, d) after 2000 steps forward followed by 2000 steps backward.

Clearly, as N increases the accuracy of the solution decreases due to the accumulation of the numerical error. For N=2000, the simulation starts to break down due to the stretching of the circle by the shearing flow that results in thin filaments only one or two cells wide which cannot be accurately represented with the current mesh resolution. As demonstrated in the forthcoming sections the amount of error is expected to decrease as the mesh resolution is increased. Compared to the Y-VOF simulation results in Rudman (1997) for the same advection test, the PLIC-VOF method implemented in this work is less accurate despite the fact that both works use the same reconstruction

algorithm. The discrepancy lies in the different advection algorithms used; an operator split method in this work and an un-split method in Rudman's work. As mentioned previously, split methods are much easier to implement but are less accurate than the un-split methods. This is confirmed by the simulation results presented herein and is consistent with other studies, i.e. Puckett and al (1995).

Table 4-1: Errors associated with the VOF advection test for shearing flow for different number of backward steps and the un-split and the split methods.

N	Split method (present work)	Un-split method (Rudman 1997)
250	4.20×10^{-3}	2.61×10^{-3}
500	8.20×10^{-3}	5.12×10^{-3}
1000	1.76×10^{-2}	8.60×10^{-3}
2000	1.02×10^{-1}	3.85×10^{-2}

4.2.2 CSF Model Tests

A common benchmark test used to validate the implementation of the CSF model is the liquid drop simulation. In the absence of viscous, gravitational, or other external forces surface tension causes a static liquid drop to become spherical (Brackbill et al., 1992). A correct implementation of the CSF model maintains the spherical shape of the drop over time. For a drop surrounded by a background fluid at zero pressure, the internal drop pressure, p_{drop}, can be evaluated by Laplace's formula,

$$p_{drop} = \sigma k = \frac{\sigma}{R} \tag{4.3}$$

where R is the drop radius (Brackbill et al., 1992). Also, as previously stated, an undesirable characteristic of the CSF method is the presence of the so-called spurious currents which can be reduced by employing a smoothed F field to perform the relevant calculations. The amplitude of these spurious currents is proportional to σ / μ_l which is equivalent to having an approximately constant value of the capillary number $Ca_s = V_s \mu_l / \sigma$ (Popinet & Zaleski, 1999). Thus, the magnitude of the spurious currents in a simulation can be approximated as

$$V_s \approx C_s \frac{\sigma}{\mu_l}, \tag{4.4}$$

with C_s = 0 being the optimum value (Popinet & Zaleski, 1999; Scardovelli & Zaleski, 1999). For this simulation, the computational domain is 0.075 m in the x and y direction with a 30 x 30 grid resolution. A circular drop with $R = 0.075/4$ m is centered at the point (0.375 m, 0.375 m). The fluid properties are ρ = 797.88 kg/m^3 and μ = 1.2x10^{-3} N.s/m^2 for the ethanol drop and ρ = 1.1768 kg/m^3 and μ = 1.0x10^{-5} N.s/m^2 for the surrounding gas. The surface tension coefficient is σ = 0.02361 N/m. Free-slip velocity boundary conditions are applied at all walls with the Van Leer limiter selected for the velocity advection scheme. For this configuration, equation (4.3) yields a pressure jump value of 1.26 N/m^2. The simulation is advanced in time for one second and was performed with the unsmoothed and the smoothed F field with ten filter passes applied. The results are shown in Figure 4.3 for both cases where C_s was found to be 6.69x10^{-5} and 2.55x10^{-4} for the unsmoothed and smoothed cases, respectively. As

expected, smoothing the F field reduces the amplitude of the spurious currents. Comparison between the theoretical pressure jump value and the estimated value for each case are presented in Figure 4.4. It is obvious that the solution of the smoothed field yields a more accurate approximation of the pressure jump than the unsmoothed field. It is important to note that the number of filter passes in this simulation is not necessarily the optimum number and is arbitrary selected only to demonstrate its effect on the accuracy of the simulation.

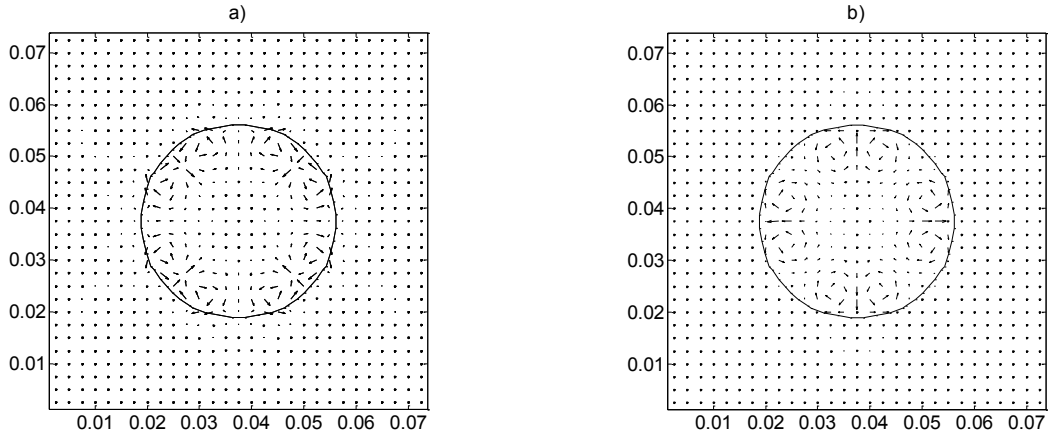

Figure 4.3: Spurious currents in a 30 x 30 grid shown for the simulation of a static liquid drop in zero gravity, a) with no filtering, and b) with ten filter passes applied. The interface is presented by the contours of $F = 1/2$.

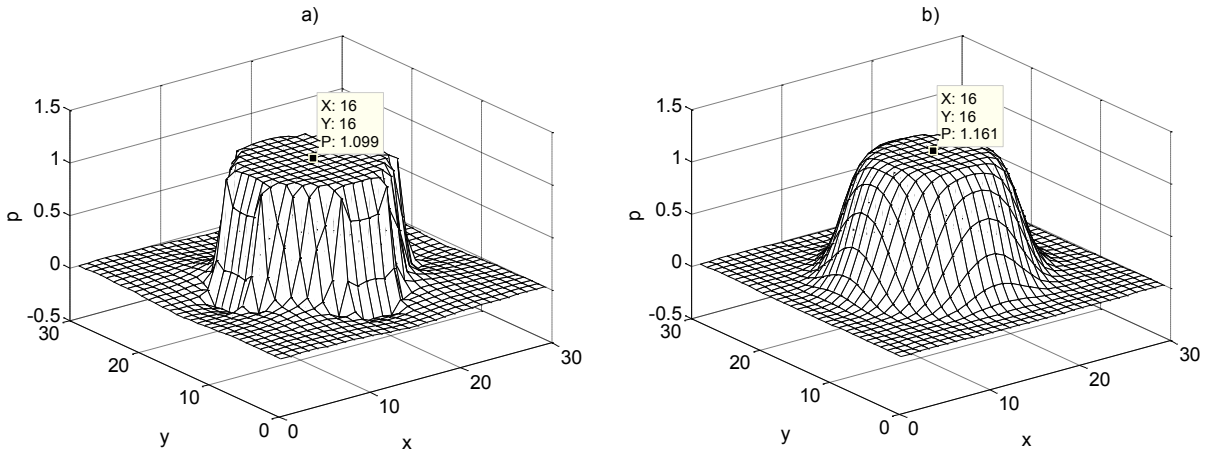

Figure 4.4: Simulation results for pressure estimation within the drop in a 30 x 30 mesh for a) unsmoothed F field, b) smoothed F field with ten filter passes. The analytical pressure is 1.25 (N/m^2).

A more realistic test case for the CSF model is one that takes into account other fluid forces such as viscous forces with the full Navier-Stokes equation being considered. To this end, the circular drop of the previous simulation is replaced by a rectangular drop as shown in Figure 4.5a. In such a case, capillary waves are induced that cause the drop surface to oscillate about its equilibrium shape, i.e. a circle (Brackbill et al., 1992). The results are shown in Figure 4.5 for simulation times, t = 0, 0.05, 0.1, 0.2, 1.0 and 2.0 seconds. As indicated by the large velocity vectors, the initial square shape of the drop results in very strong surface forces at the high-curvature corners, setting the drop into oscillation. As the solution is advanced in time, oscillations are eventually damped and the drop approaches its circular equilibrium shape. It is easy to verify that the 0.40 second oscillation period reported in Brackbill et al. (1992) is recovered in this simulation.

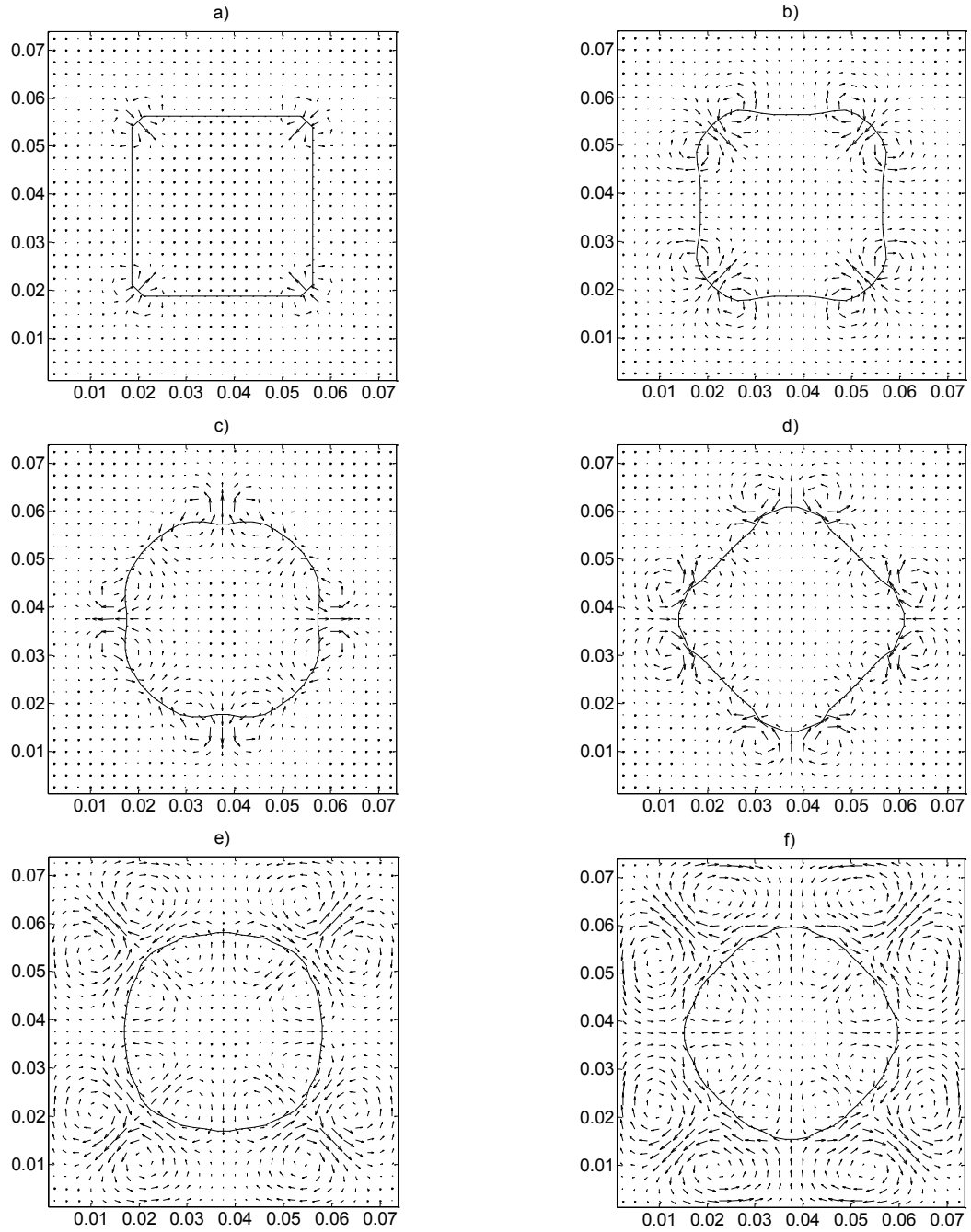

Figure 4.5: Simulation results for initially-square ethanol drop oscillation in zero gravity. The velocity field and the interface, defined by contours of $F = 1/2$, are shown at times of a) 0.0, b) 0.05, c) 0.1, d) 0.2, e) 1.0, f) 2.0 seconds.

4.3 Rayleigh-Taylor Instability

Rayleigh-Taylor instability is a common test problem for methods intended to simulate multiphase flows and is relatively easy to set up. It consists of a heavy fluid that is supported against gravity by a lighter fluid and is then given a small perturbation at the interface boundary. The perturbation grows exponentially in time with the heavy fluid falling down in a relatively narrow spike while the lighter fluid rises upward as a large bubble (Brackbill et al., 1992; Tryggvason et al., 1998).

The case studied here is similar to the setup suggested in Puckett et al. (1997); the air/helium Rayleigh-Taylor instability is considered in a 0.01m x 0.04m domain with a 64 x 256 mesh resolution. The top fluid is air with ρ = 1.255 kg/m^3 and μ = 1.77625x10^{-5} N.s/m^2 and the lower fluid is helium with ρ = 0.1694 kg/m^3 and μ = 1.941x10^{-5} N.s/m^2. The surface tension coefficient is 0.0 N/m. Velocity boundary conditions are no-slip at the top and bottom wall and free-slip at the left and right wall. The Van Leer limiter is selected for the velocity advection scheme. The interface is initially a sine wave with amplitude 0.05 m. Figure 4.6 compares the simulation results of the present work and the results obtained in Puckett et al. (1996) for different flow times. The predicted interface shape is essentially identical for both tests.

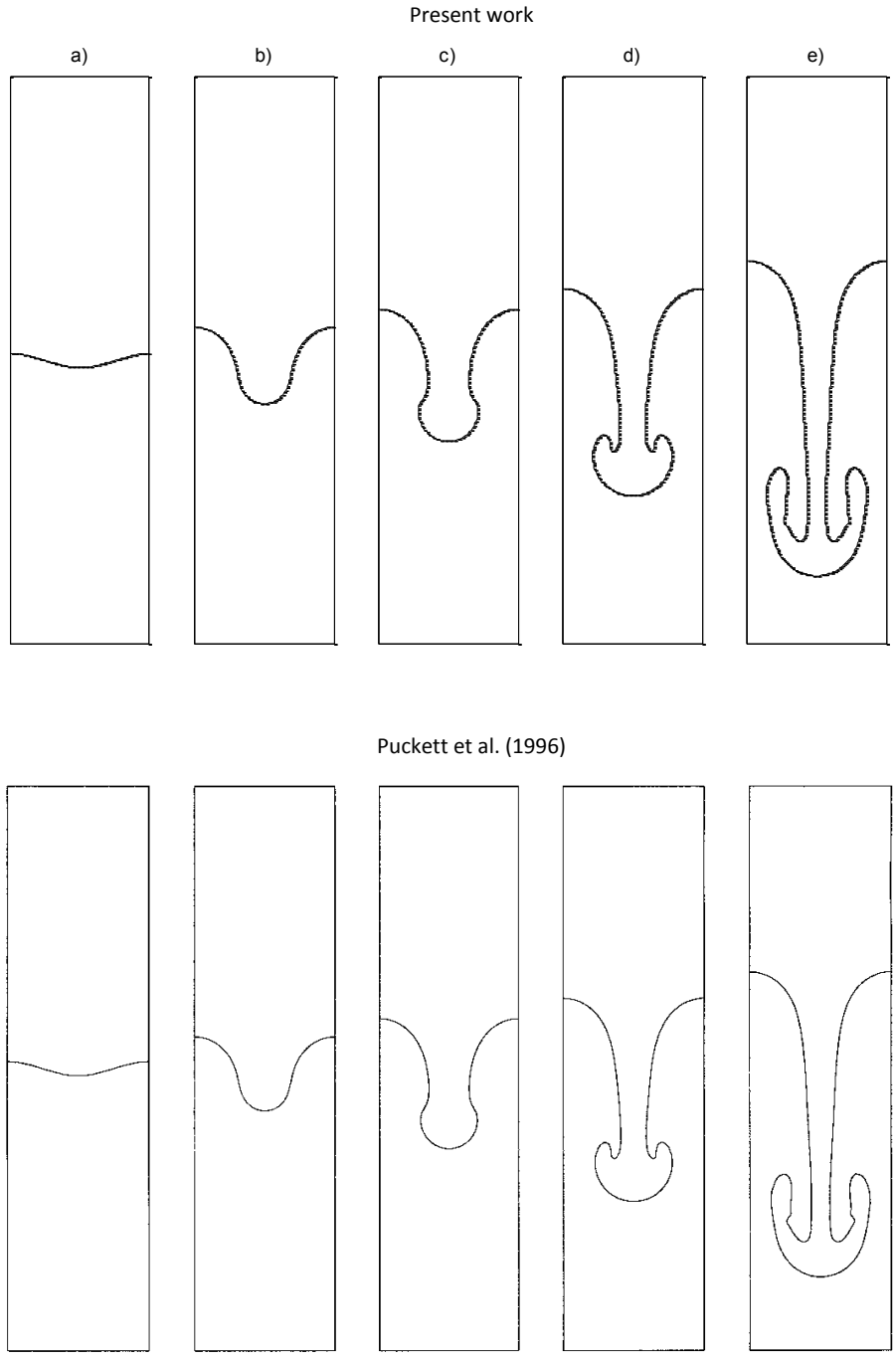

Figure 4.6: Comparison of the simulation results of the Rayleigh-Taylor instability for air-helium in a 64 x 256 mesh at times, a) 0.0, b) 0.047, c) 0.66, d) 0.88 and e) 0.118 seconds.

4.4 Grid Refinement Test

In order to demonstrate the convergence properties of the spatial discretization of the numerical scheme a grid refinement test is performed on the Rayleigh-Taylor instability test as outlined in Puckett et al. (1996). However, to allow for all the Navier-Stokes equation terms to contribute to the motion of the fluid, the setup suggested in Tryggvason (1998) is used instead. Here, the lower fluid properties are ρ = 1 kg/m^3 and μ = 0.01 N.s/m^2, the top fluid properties are ρ = 5 kg/m^3 and μ = 0.1 N.s/m^2 and the surface tension coefficient is 0.015 N/m. The computational domain stretches 1 m in the x direction and 3 m in the y direction. Again, velocity boundary conditions are no-slip at the top and bottom wall and free-slip at the left and right wall with the Van Leer limiter selected for the velocity advection scheme. The instability is initiated with a perturbation of the form $y = 0.05\cos(2\pi x)$ applied at the interface. The results are computed to time 0.120 seconds. The solution is computed for three different mesh resolutions, 8x24, 32x96 and 128x384. Subsequently, solutions on meshes of adjacent resolution are compared by averaging the fine solution onto the coarse one and approximating the error on the coarser mesh as follows:

$$E = \frac{\sum_{i,j=1}^{N_{mesh}} \left| \overline{S} - S \right|}{N_{mesh}}, \tag{4.5}$$

where S is the solution of the coarser mesh and $\overline{S}$ is the solution of the finer mesh averaged onto the coarser mesh (Ferziger & Peric, 2002). Finally, for constant ratio mesh refinements the observed order of the truncation rate decay can be calculated as

$$\bar{p} = \ln\left(\frac{E_{32}}{E_{21}}\right) \Big/ \ln(rr), \tag{4.6}$$

where E_{32} is the error associated with the medium mesh compared to the fine mesh, E_{21} is the error associated with the coarse mesh compared to the medium mesh and rr is the refinement ratio which is 4 in this case (Versteeg & Malalasekra, 2007). For this problem, $\bar{p}$ was found to be 0.9455 for the volume fraction field and 0.5 for the velocity field. This indicates that the VOF advection scheme performs near its formal first to second-order accuracy while the momentum transport scheme performs at an order half of its formal convergence rate expected for a discontinuous velocity field. The above test was repeated by setting the surface tension coefficient to zero and equating the viscosity of both fluids (μ = 0.1 N.s/m^2). Doing so effectively makes the velocity field continuously differentiable across the entire domain. For this configuration, $\bar{p}$ was found to be 1.05 for the volume fraction field and 0.94 for the velocity field. These results are consistent with the trends obtained in Puckett et al. (1996) for continuous and discontinuous velocity fields where the nearly second-order convergence rates obtained for the continuous test were reduced to convergence rates below 1 for the discontinuous test. The inability of the method to perform beyond first-order for the velocity field could be attributed to the first-order approximation of the volume fraction field that is in turn used to calculate the fluid properties appearing in the momentum transport equation. This implies that the overall accuracy of the method cannot exceed that of the volume fraction field. Grid refinement test results are shown in Figure 4.6

and Figure 4.7 for the discontinuous velocity field and the continuous velocity field, respectively. Even for the relatively coarse mesh development of the expected mushroom cap can be observed in both cases.

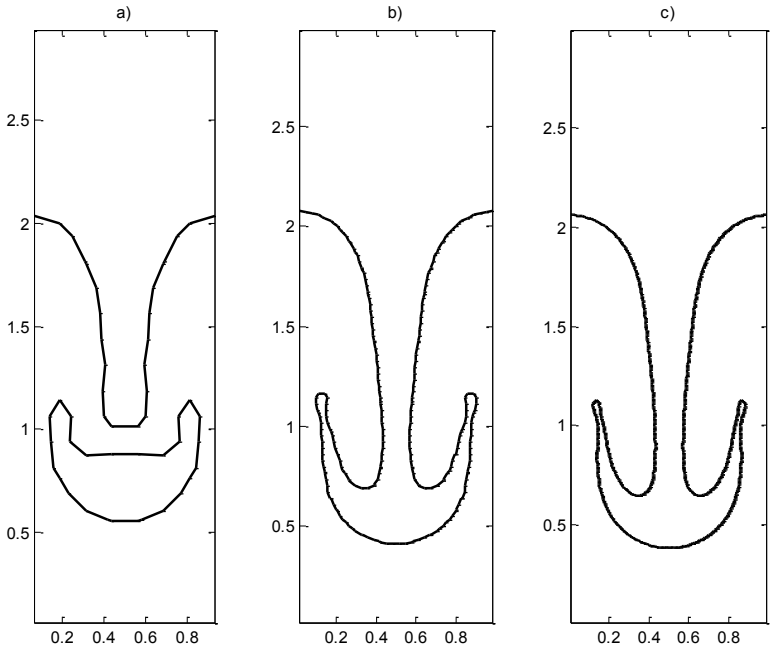

Figure 4.7: Grid refinement test results for the discontinuous velocity field shown at time, t = 1 for different mesh resolutions, a) 8x24, b) 32x96, and c) 128x384.

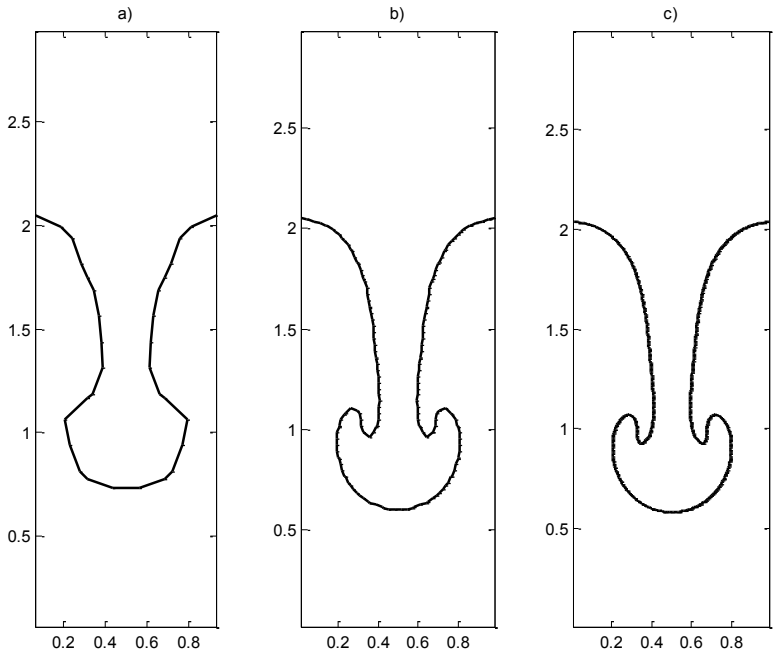

Figure 4.8: Grid refinement test results for the continuous velocity field shown at time, t = 1 for different mesh resolutions, a) 8x24, b) 32x96, and c) 128x384.

4.5 Summary

Standard benchmark tests were performed on the implementation of the VOF and the CSF surface tension model. The ability of the VOF method in handling fluid merging and breakup and the affects of smoothing on the accuracy of the CSF model were highlighted. This was followed by the simulation of Rayleigh-Taylor instability to validate the overall numerical scheme. It was verified that the simulation results compared well with the expected results found in the literature. Finally, a grid refinement test was performed to obtain a quantitative assessment of the order of accuracy of the numerical scheme; it was found that the overall accuracy of the algorithm cannot exceed that of the first-order accurate VOF model as a result of the coupling between the VOF advection scheme and the material properties that appear in the momentum transport.

Chapter 5

Applications

5.1 Overview

In this chapter two practical cases are considered for the comparison of numerical predictions against the experimental data or other numerical results found in the literature. In the first test, the numerical simulation of a bubble rising and deforming in quiescent viscous liquid is performed to measure the performance of the algorithm for the computation of low Reynolds number flows ($Re < 20$). In the second test, a more challenging case involving the flow of a falling liquid film exposed to a forced disturbance is simulated to measure the ability of the numerical scheme in capturing the physics of small scale fluid mechanics problems for moderate Reynolds number flows (Re<200). Both cases involve very large ratios of fluid properties across the thin interface which can complicate the numerical simulation due to instabilities and low accuracy near the interface. Although these cases are not particularly difficult to set up for a numerical simulation, they could reveal useful information on the scope of the physical problems that a numerical method can be used for.

5.1.1 Application 1: Bubble Rising in a Viscous Liquid

5.1.1.1 Introduction

A fundamental understanding of the behaviour of a bubble rising and deforming in a quiescent viscous liquid is crucial in various applications of multiphase flows ranging from the rise of steam in boiler tubes to gas bubbles in oil wells (C. Li, Garimella, Reizes, & Leonardi, 1999). Thanks to the large body of numerical and experimental data that exist in the literature, the rise of a bubble can also serve as an ideal validation test for numerical schemes. Bhaga and Weber (1981) analysed a large body of experimental data on shape and rise velocities of bubbles in quiescent viscous liquids and presented their findings in a diagram similar to the one depicted in Figure 5.1. van Sint Annaland et al. (2005) and Li et al. (1999) used the VOF method and Hua et al. (2008) used the front-tracking method to obtained numerical approximations for the rise of a bubble.

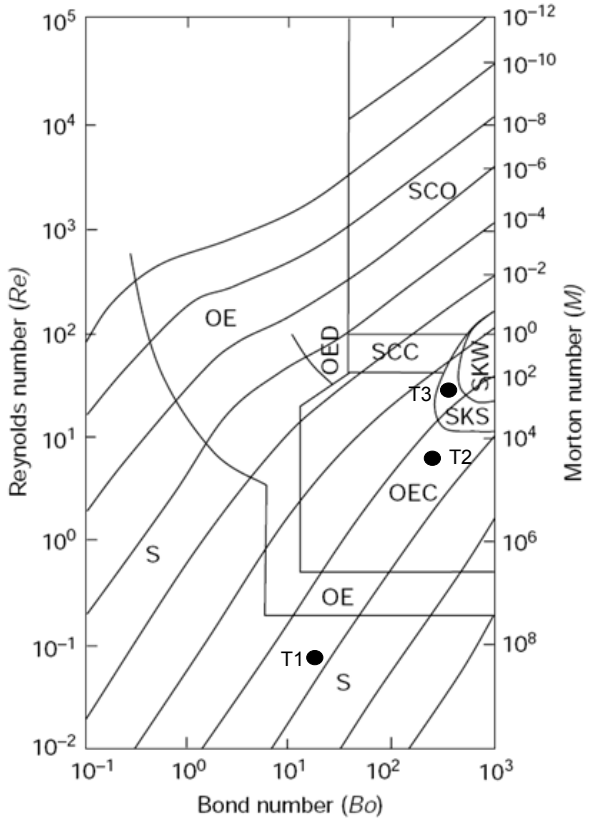

Figure 5.1: Regime map of experimentally observed rising bubble shape as reported by Bhaga and Weber (1981). Bubble shape regions are identified as spherical (S), oblate ellipsoid (OE), oblate ellipsoid wobbling disk (OED), oblate ellipsoidal cap (OEC), spherical cap with close steady wake (SCC), spherical cap with open unsteady wake (SCO), skirted with smooth, steady skirt (SKS) and skirted with wavy, unsteady skirt (SKW). (Yeoh & Tu, 2010)

5.1.1.2 Governing Equations

The problem of a bubble rising in a liquid studied herein is considered to be a multi-fluid, laminar, isothermal flow with incompressible and immiscible Newtonian fluids. The governing equations are the Navier-Stokes and continuity equations (equation 2.21 and 2.22) that can be non-dimensionalized by introducing the following dimensionless characteristic variables:

$$\tilde{V} = \frac{}{\sqrt{\|\vec{g}_{\|}\| a}} \qquad \sqrt{\frac{\|\vec{g}_{\|}\|}{d}} t, \tilde{} \qquad \frac{}{\rho_L} \tilde{} \qquad \frac{}{\rho_L \|\vec{g}_{\|}\| a} \qquad \frac{}{\mu_L} \qquad \frac{}{\|\vec{g}_{\|}\|} \tag{5.1}$$

where d is the characteristic length of the problem and is selected to be the diameter of the initial bubble, $\|\vec{g}_{\|}$ is the magnitude of $\vec{g}$ and subscript L specifies the liquid phase properties (Hua, Stene, & Lin, 2008). Thus, omitting the superscript ~ for convenience, the non-dimentionilized Navier-Stokes and continuity equations for the motion of a bubble rising in a liquid can be re-expressed as

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) = -\frac{}{\rho} \nabla p + \frac{}{Ar} \frac{}{\rho} \nabla \cdot \mu \left[\left(\nabla \vec{V} \right) + \left(\nabla \vec{V} \right) \right] + \vec{g} + \frac{}{Bo} \frac{}{\rho} \sigma k \nabla F, \tag{5.2}$$

$$\nabla \cdot \vec{V} = 0. \tag{5.3}$$

The relevant non-dimensional numbers are the Archimedes number defined as

$$Ar = \frac{\rho_L g^{1/2} d^{3/2}}{\mu_L}, \tag{5.4}$$

the Bond or Eotvos number defined as

$$Bo = \frac{\rho_L g d^2}{\sigma}. \tag{5.5}$$

In experimental work the Reynolds number is often used to describe the flow regime in conjunction with the Bond and Morton number. Denoting the measured bubble terminal rising velocity by U_{∞}, the Reynolds number is defined as

$$Re = \frac{\rho_L d U_\infty}{\mu_L}, \tag{5.6}$$

and the Morton number is defined as

$$M = \frac{g \mu_L^4}{\rho_L \sigma^3}. \tag{5.7}$$

5.1.1.3 Simulation Setup

For the purpose of this work, 3 test cases are selected as defined in Table 5-1. Following the discussion presented in sin Von Annaland et al. (2005) and Hua et al. (2008) regarding the wall effects on the terminal rising velocity of the bubble the width of the computation domain in the x direction is selected such that it is at least 4 times the initial diameter of the bubble. The computational domain is therefore, $4d$ and $6d$ in the x and y direction, respectively. An 80 x 120 mesh was selected to carry out the simulation such that the initial bubble is presented by 20 cells in each direction. Increasing the grid resolution beyond this did not affect the simulation results. The liquid to gas density and viscosity ratio in all the test cases are fixed to 1000 and 100, respectively, in order to mimic gas-liquid flows usually involved in these flows. Free-slip velocity boundary conditions are applied to all the confining walls to allow for an easy cycle of the fluid near the lateral boundaries. The bubble is initially centered at point (x,y) = (2,1.5). Simulations were terminated before the bubble approaches the top wall at time τ = 4.

Table 5-1: Physical parameters related to the test cases simulated in this work.

Parameter	Test case 1 (T1)	Test case 2 (T2)	Test case 3 (T3)
Bo	17.7	243	339
M	711	266	43.1
Re	0.232	7.77	18.3

5.1.1.4 Simulation Results and Discussions

The final bubble shapes are shown in Figure 5.2 for each test case and compare well with the simulation results of Hua et al. (2008) and the experimental results of Bhaga and Weber (1981). Despite the two-dimensional setup of the simulation, the method seems to be capable of capturing the main physical characteristics of each test case such as the spherical bubble shape in test case 1, the oblate ellipsoidal cap in test case 2 and the steady wakes in test case 3. Also, as expected the bubble deforms more as the Bonds number increases.

Instantaneous bubble rising velocities were measures for each case and are presented in Figure 5.3 as a function of τ. For all the cases the bubble reaches its terminal velocity after $\tau = 3$.This allows for the simulation Reynolds number to be calculated for each test case as per equation (5.6). The experimental bubble Reynolds number can be determined for each test case from Figure 5.1 where test cases 1 to 3 are identified as T1, T2 and T3, respectively. The simulation results are summarized in

Table 5-2 and are compared to the experimental values. It can be seen that the results compare less favourably with the experimental values as the Reynolds number

increases. This could be attributed to the lack of three-dimensional flow dynamics in a two-dimensional simulation.

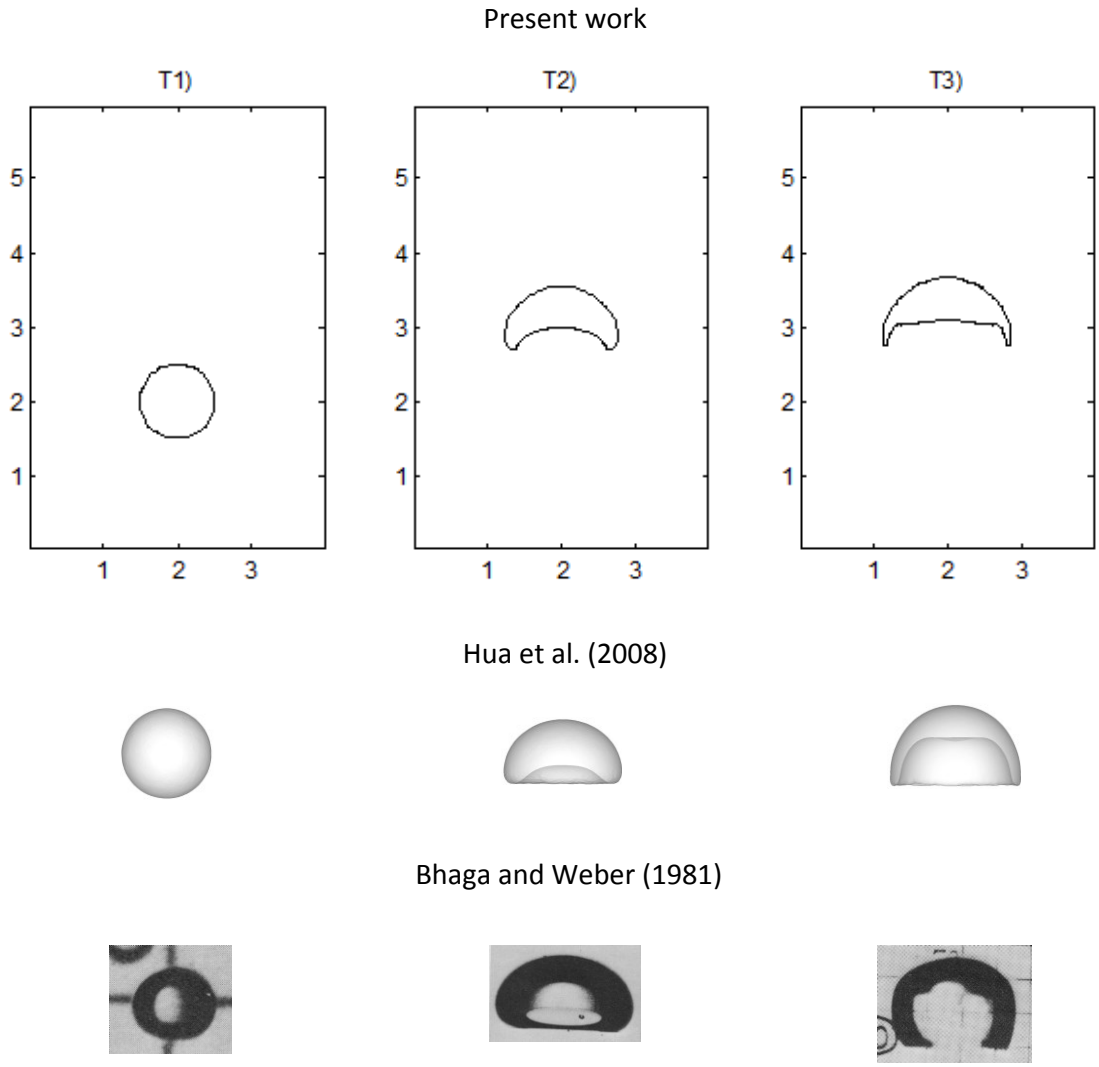

Figure 5.2: Comparison of the final bubble shapes with other numerical and experimental results for different test cases. Contours of $F = 0.5$ are shown for the results of the present work.

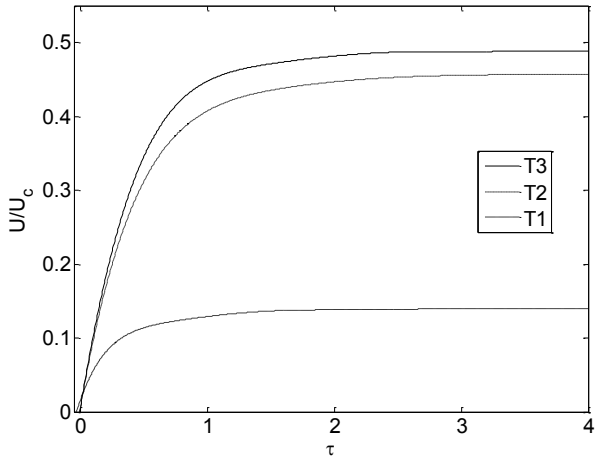

Figure 5.3: Instantaneous bubble rising velocity normalized for $U_c = \sqrt{\|\vec{g}\| a}$, as predicted numerically for different test cases.

Table 5-2: Comparison of the terminal velocities found experimentally by Bhaga and Weber (1981) and numerically by Hua et al. (2008) to the predicted results of the present work.

Test Case	Re			%Deviation from experiment
	Experimental (Bhaga and Weber (1981))	3D Numerical (Hua et al. (2008))	2D Numerical (Present work)	
T1	0.232	0.182	0.244	5.17
T2	7.77	7.605	7.000	9.90
T3	18.30	17.758	15.220	16.83

5.1.2 Application 2: Falling Liquid Film

5.1.2.1 Introduction

Liquid film falling down an inclined plane is observed in a wide variety of naturally occurring phenomena and is also used in many industrial processes and engineering devices such as condensers, evaporators and chemical reactors (Kunugi & Kino, 2005). This kind of flow exhibits a rich variety of surface wave dynamics and therefore has been the subject of many theoretical, experimental and numerical studies. Past studies have determined that the instability of thin film flow is dominated by gravity, viscous forces and surface tension effects (Gao et al., 2003) which amount for an ideal test case to validate a multiphase flow numerical method and assess its performance. Compared to the rising bubble test, simulation results for this problem are more difficult to obtain and are a better indication of the true abilities and limitations of the algorithm since the small scales of the problem make all the terms of the Navier-Stokes equation equally important to the motion of the thin film. Results obtained by previous numerical simulations (Gao et al. (2003) and Kunugi and Kino (2005)) as well as experimental data and observations (Kapitza and Kapitza (1964) and Nosoko et al. (1995)) can be used to evaluate the integrity and precision of the numerical algorithm.

5.1.2.2 Governing Equations

The flow of the falling liquid film studied in this work is considered to be a multiphase, laminar, isothermal flow with incompressible and immiscible Newtonian fluids. The governing equations are the Navier-Stokes and continuity equations presented in

chapter 2 (equations (2.21) and (2.22)). These equations can be non-dimensionalized by introducing the following characteristic dimensionless parameters:

$$\tilde{V} = \frac{\vec{V}}{u_c}, \quad \tilde{t} = \frac{t}{t_c}, \quad \tilde{\rho} = \frac{\rho}{\rho_L}, \quad \tilde{p} = \frac{p}{\rho_L u_c^2}, \quad \tilde{\mu} = \frac{\mu}{\mu_L} \tag{5.8}$$

where u_c is the characteristic velocity of the problem and is set to the average velocity of the fully developed flow defined as (Gao et al. (2003))

$$u_c = \frac{\rho_L g h_c^2 \sin\gamma}{3\mu_L}, \tag{5.9}$$

h_c is the characteristic length of the problem and is selected to be the initial flat film thickness, t_c is the characteristic time in second defined as h_c/u_c, γ is the inclination angle of the plate and subscript L refers to the fluid properties.

Therefore, omitting the superscript ~ for convenience, the non-dimensional Navier-Stokes and continuity equations can be re-expressed as

$$\frac{\partial \vec{V}}{\partial t} + \nabla\cdot(\vec{V}\vec{V}) = -\frac{1}{\rho}\nabla p + \frac{1}{Re}\frac{1}{\rho}\nabla\cdot\mu\left[\left(\nabla\vec{V}\right) + \left(\nabla\vec{V}\right)^T\right] + \frac{1}{Fr^2}\vec{e} + \frac{1}{We}\frac{1}{\rho}\sigma k\nabla F, \tag{5.10}$$

$$\nabla\cdot\vec{V} = 0, \tag{5.11}$$

where $\hat{e} = (\sin\gamma, \cos\gamma)$ is the unit gravity vector.

The relevant dimensionless numbers are the Reynolds number

$$Re = \frac{\rho_L u_c h_c}{\mu_L}, \tag{5.12}$$

the Weber number

$$We = \frac{\sigma}{\rho_L u_c^2 h_c}, \tag{5.13}$$

and the Froude number

$$Fr = \sqrt{\frac{u_c^2}{g h_c}}. \tag{5.14}$$

Normalization by h_c allows for the Froude number to be expressed as

$$Fr = \sqrt{Re/3}. \tag{5.15}$$

Finally, it can be shown that the velocity profile of the fully developed flow normalized for h_c and u_c can be expressed as

$$u_{fd} = \frac{3}{2}\left(2Y - Y^2\right), \tag{5.16}$$

where $Y = y / h_c$ (Gao et al. (2003)).

5.1.2.3 Simulation Setup

One of the main difficulties associated with numerical and experimental study of falling liquid film flows is the typical length scale that is required to allow for the surface waves to evolve into steady-state periodic waves. Past experiments, such as Nosoko et al.

(1995), have shown that in case of a vertical falling film, with an inclination angle of 90°, the running length can be minimized to 10 to 13 centimetre as opposed to 150 centimetres needed for an inclination angle of 4° to 10° reported in Liu and Gollub (1994). Moreover, despite the fact that the experimental evidence indicates that two-dimensional regular wave pattern can result from natural disturbances, obtaining regular wavy surface has proven to be difficult without a periodic forcing at the inlet (Gao et al., 2003). Thus, a feasible numerical simulation of a falling film flow requires an inclination angle of $\gamma = 90^{\circ}$ with a periodic flow rate perturbation at the inlet as depicted in Figure 5.4.

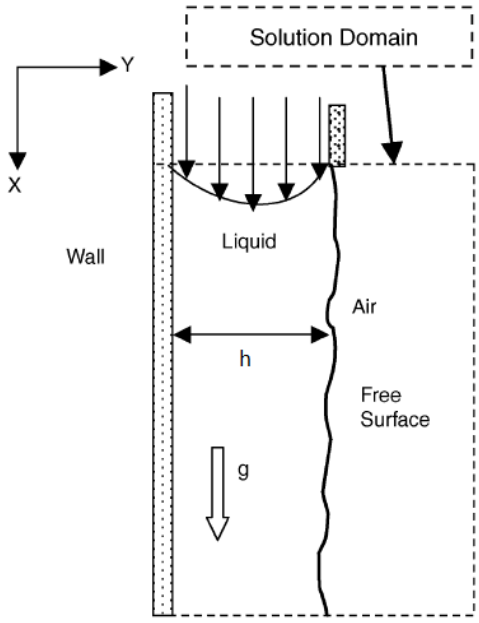

Figure 5.4: Numerical configuration of the vertical falling film flow. (Kunugi and Kino, 2005)

The simulation configuration described herein is similar to the ones conducted in Gao et al. (2003). To set up the simulation of the falling film flow, a computational domain

stretching 600 h_c or 500 h_c in the x direction and 20 h_c in the y direction is defined with the xy axis located at the intersection of the plate and the left hand wall and gravity acting along the positive x direction (Figure 5.4). The velocity boundary conditions are no-slip at the top and bottom walls and inflow and outflow at the left and right boundaries, respectively. The entire computational domain is initialized to the fully developed flow velocity profile given by equation (5.16). This of course is the same velocity profile that one expects to obtain if the computation domain was initially void of any fluid and the steady inflow was set to let the liquid flow over the plate until a steady state is reached. However, that would require extra computational time as well as an extended computational domain to allow for the flow to become fully developed. The same inflow velocity profile is fixed over the flat film thickness throughout the entire simulation with a time dependant perturbation applied as follows

$$u_{in} = \frac{3}{2}\left(2Y - Y^2\right)u_p, \tag{5.17}$$

where u_p introduces the desired level of monochromatic time varying perturbation according to the following relation

$$u_p = 1 + \varepsilon \sin\left(2\pi f_p t\right). \tag{5.18}$$

Here f_p is the forcing frequency in Hertz, t is time in seconds and ε is the dimensionless disturbance magnitude.

Two test cases are defined for the simulation of the vertical falling liquid film with water as the working fluid surrounded by air. The values of the relevant dimensionless parameters for each test case are defined in Table 5-3. Material properties and other parameters used for this simulation are listed in Table 5-4. As was the case in Gao et al. (2003), the optimum mesh resolution for the first test case was determined to be 1500 in the x direction and 20 in the y direction. However, for the second test case with the higher Reynolds number, a 800 x 20 mesh resolution was found to yield the best results even compared to finer mesh resolution. Also, two passes of the filter defined in equation (3.30) was applied to for the simulation of the second test case. Simulations were advanced in time for t / t_c = 450.

Table 5-3: Relevant dimensionless parameters used for the falling liquid film simulation.

Test case	Re	We	f_p (Hz)	ε
TF1	20.1	33.5	27	0.05
TF2	69	4.2	30	0.05

Table 5-4: Material properties and other test parameters used in the falling liquid film simulation.

ρ_L (kg/m^3)	ρ_g (kg/m^3)	μ_L (N.s/m^2)	μ_g (N.s/m^2)	σ (N/m)	g (m/s^2)
998.2	1.225	9.87419 x 10^{-4}	1.8615 x 10^{-5}	0.0722	9.81

5.1.2.4 Simulation Results and Discussion

The predicted instantaneous film thicknesses obtained from the simulations are shown in Figure 5.5 and Figure 5.6 for test case TF1 and TF2, respectively. It can be seen that

for both cases the small disturbances introduced at the inlet propagate into the domain to form finite-amplitude nearly stationary periodic waves at the end of the simulation. The stationary wave shape obtained in both tests are comparable to the wave shapes obtained in Gao et al. (2003) for the same tests although for case TF1 the waves obtained in this work seem to penetrate deeper into the fluid side resulting in slightly taller waves. Nonetheless, as illustrated in Figure 5.7, the wave shapes obtained in this work compare well with the experimental results reported by Kapitza and Kapitza (1964) and the simulation results of Gao et al. (2003). For test case TF2 the wave shapes obtained match closely to those obtained in Gao et al. (2003) although the finer mesh used in Gao et al. (2003) seems to better resolve the capillary waves leading the solitary waves.

It should be noted that, contrary to the simulations performed in this work, the algorithm used in Gao et al. (2003) ignores the dynamics of the gas phase and treats it as a "void". This in practice, leads to a more stable algorithm for the simulation of free-surface flows but requires complex velocity and volume fraction boundary conditions to be imposed on the interface and limits the code to the simulation of free-surface flows only.

To better assess the accuracy of the simulation results, the simulation wave velocities, u_w, were calculated for each test case by measuring the distance traveled by a wave over a given time interval as shown in Figure 5.8 and Figure 5.9 for test case TF1 and TF2, respectively. The corresponding wavelengths, λ, were also determined form the

same figures. These values were then compared to the values obtained using the following equation derived by Nosoko et al. (1995) from the experimental data analysis:

$$u_w = 1.87We^{-0.06}\,Re^{-0.293}\,\lambda^{0.31}. \tag{5.19}$$

The results are summarized in

Table 5-5 and confirm the reasonable accuracy of the simulation results obtained in this work.

Table 5-5: Comparison of the simulation wave velocities with the experimental data.

Test case	Numerical (Present work)		Experimental (Nosoko et al. (1995))	%Deviation From experiment
	λ	u_w	u_w	
TF1	50	2.24	2.11	5.9
TF2	58.5	1.87	1.75	6.8

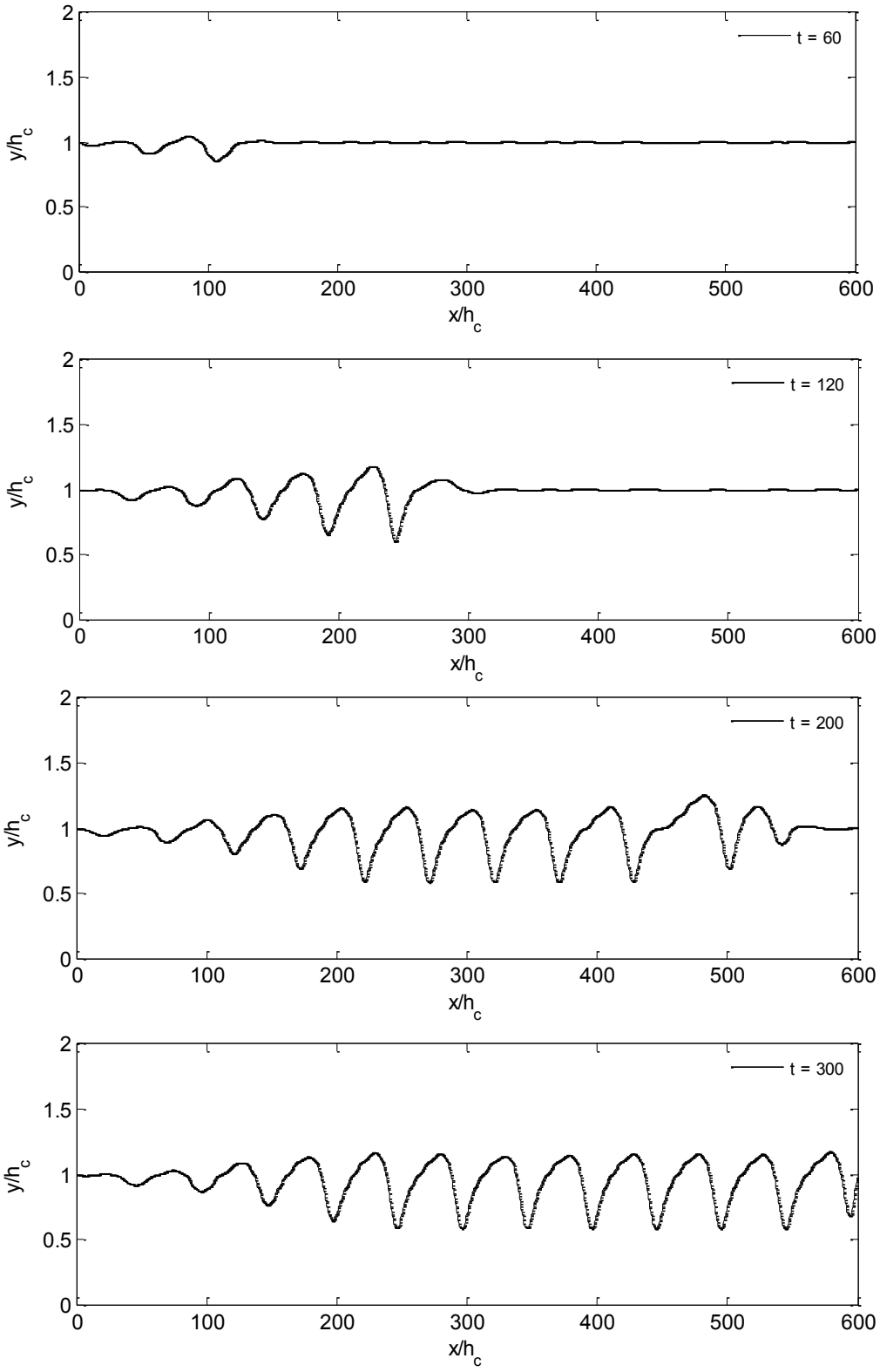

Figure 5.5: Instantaneous film thickness for test case TF1. Contours of $F = 0.5$ are shown.

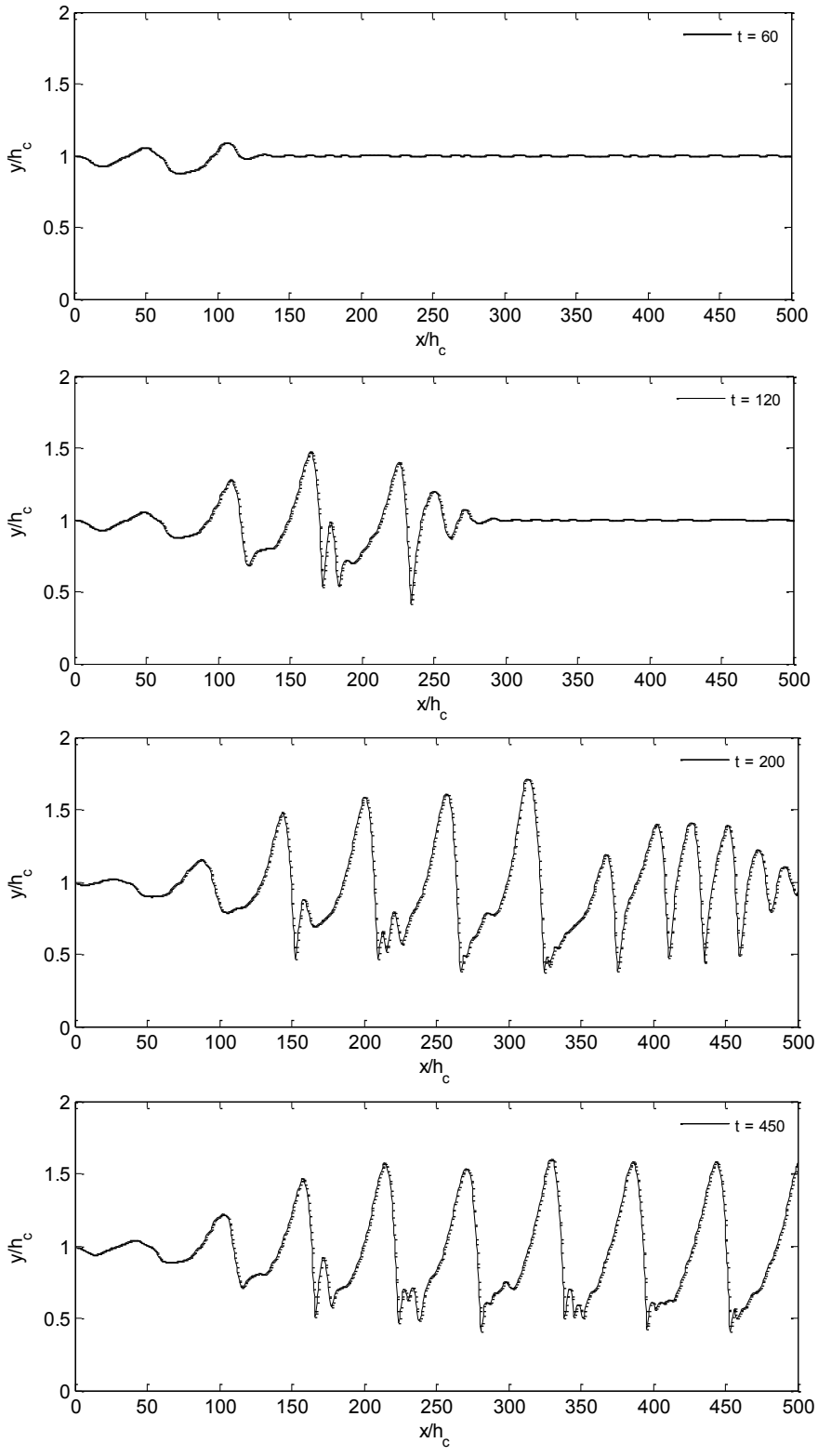

Figure 5.6: Instantaneous film thickness for test case TF2. Contours of $F = 0.5$ are shown.

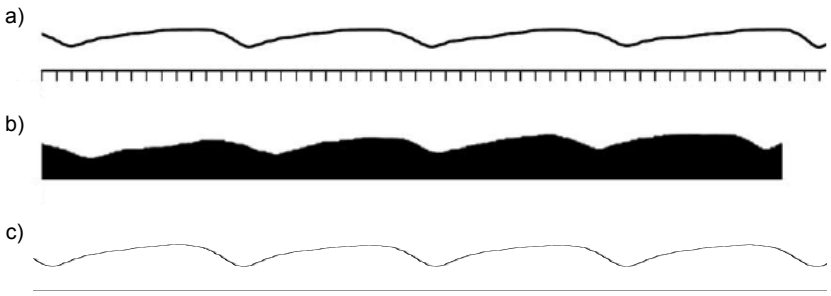

Figure 5.7: Comparison of wave shapes taken from a) Gao et al. (2003), b) Kapitza and Kapitza (1964) shadowgraph, and c) the present work, for test case TF1.

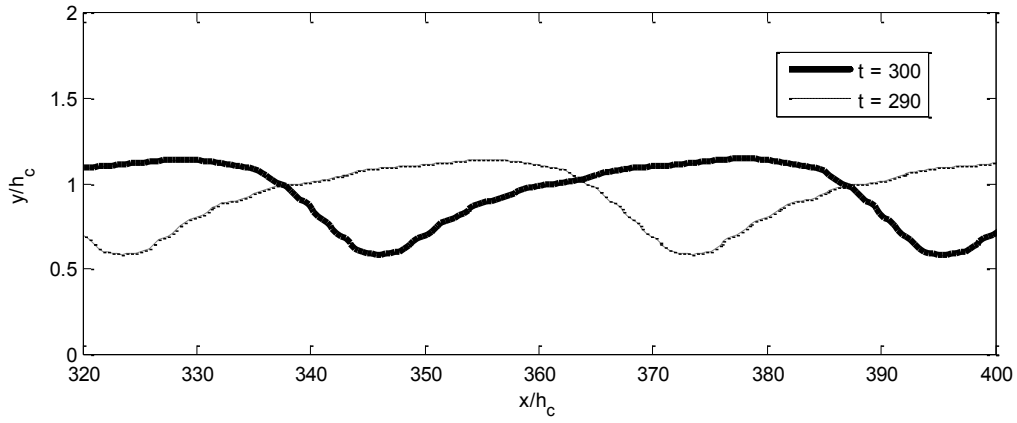

Figure 5.8: Two trains of stationary waves used to determine λ and u_w for test case TF1.

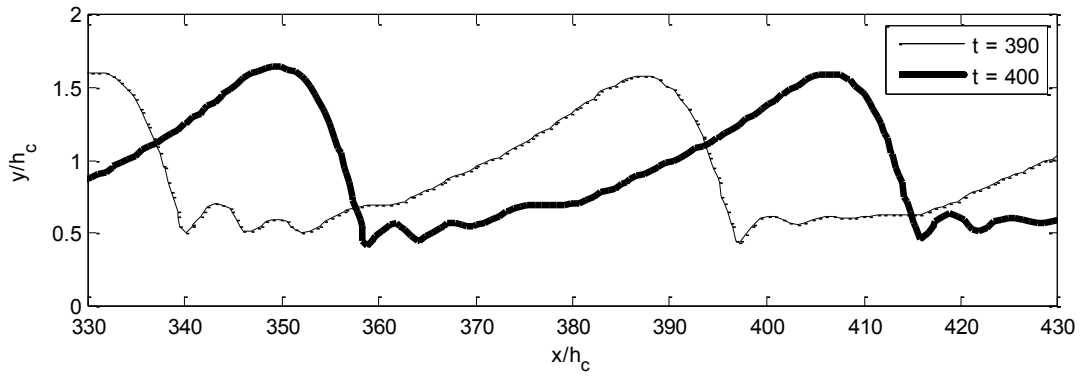

Figure 5.9: Two trains of stationary waves used to determine λ and u_w for test case TF2.

5.2 Summary

Applicability to practical physical and engineering problems was verified over a wide range of test conditions by using the numerical method for the simulation of the rise of a bubble in a liquid and the falling liquid film flow. The simulation results for both cases compared well with the past experimental and numerical results. For the rising bubble case the model successfully predicted the shape and the terminal rise velocity of the bubble but appeared to suffer from the lack of three-dimensional flow dynamics at higher Reynolds numbers. The simulation of the falling liquid film proved to be more challenging especially for the higher Reynolds number case where accurate results could only be obtained for a narrow range of mesh resolutions. However, once properly configured, the algorithm was capable of capturing the complex wave dynamics of the falling liquid film with a reasonable accuracy.

Chapter 6

Conclusion

6.1 Summary and Contributions

Following a review of the available methods for modeling the interfacial multiphase flows, a robust and efficient, yet simple numerical method for the simulation of these flows was presented. The one-fluid model and the VOF method were selected as the methods of choice due to their efficiency and simplicity, coupled with their applicability to a vast spectrum of multiphase problems. CSF model was selected to include the surface tension effects as a body force in the momentum transport, eliminating the need for an explicit treatment of the interfacial boundary conditions. The corresponding governing equations and their numerical approximation were presented in detail followed by a brief discussion on the merits of MATLAB as a favourable programming environment especially due to its built-in parallel computing capabilities. The final code was validated through various benchmark tests through which implementation of the VOF method and the CSF model was verified. Grid refinement tests revealed that for continuous velocity fields the model performs near its formal first-order accuracy while for discontinuous velocity fields the order of accuracy was reduced by nearly 1/3. This also verified that the overall order of accuracy of the numerical model cannot exceed the first-order accuracy of the VOF advection and the CSF model. Applicability to real fluid mechanics problems was studied for two test cases where capturing the true

physical phenomena, rather than the order of accuracy, was of prime interest. Application to the rise of a bubble in quiescent fluid was relatively straight forward and yielded reasonably accurate results when compared to the experimental data. The simulation seemed to suffer from the lack of three-dimensional flow effects at higher Reynolds numbers. Numerical study of the vertical falling liquid film proved to be more challenging as the high Reynolds number case needed a careful selection of the mesh resolution in order to yield physically meaningful simulation results. The final results, however, compared well with the results of the experimental and numerical studies found in the literature.

The relatively simple setup required to simulate a versatile array of physical problems with the same code underlined the simplicity and robustness of the selected numerical method. The results obtained, especially for the falling liquid film problem, were on a par with the other numerical results that used more complicated algorithms that are more suitable for the simulation of free-surface flows but are less flexible for general multiphase flow simulations where the dynamics of both fluids need to be studied.

Implementation in MATLAB proved to be quite advantageous for the simulation of the falling liquid film flow where the parallel computing capabilities of MATLAB reduced the simulation times significantly. No extra code was needed for post processing since the built-in tools available in MATLAB were used to present the data and visualize the findings. Debugging, testing and optimizing the code was also facilitated using many of the great tools that MATLAB provides. The complete algorithm was implemented in

MATLAB in less than 200 lines compared to the 2000+ lines needed for the FORTRAN implementation of the SOLA-VOF (Nichols et al. (1980)) that has no parallel computing capabilities and uses the simpler VOF-SLIC algorithm. This makes the final code more accessible and easier to maintain and improve in the future.

6.2 Future Directions

Given the adequate performance of the numerical scheme selected in this work and the benefits of the parallel computing capabilities offered by MATLAB, the numerical method presented herein can be used as a baseline for future studies and developments. Extension to three dimensions seems to be the logical step forward as all the numerical schemes selected have already been applied in three dimensions and benefit from a vast collection of literature dedicated to this subject. In the context of the two-dimensional simulations, the accuracy of the final results can be improved by employing higher order VOF and CSF schemes since the overall accuracy of the model was found to be limited by their order of accuracy. Finally, due to its finite-volume nature, the code can be used as a starting point for developing more advanced codes that are capable of simulating flows in complex geometries, although that might require a purely algebraic approach to the VOF method as opposed to the geometric method used in this work.

References

Aulisa, E., Manservisi, S., & Scardovelli, R. (2003). A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. *Journal of Computational Physics, 188*(2), 611-639.

Balachandar, S. (2007). Structured grid methods for solid particles. In A. Prosperetti, & G. Tryggvason (Eds.), *Computational methods for multiphase flow* (pp. 78-112). Cambridge: Cambridge University Press.

Begum, R., & Abdul Basit, M. (2008). Lattice boltzmann method and its applications to fluid flow problems. *European Journal of Scientific Research, 22*(2), 216-231.

Bhaga, D., & Weber, M. E. (1981). Bubbles in viscous liquids: Shapes, wakes and velocities. *Journal of Fluid Mechanics, 105*, 61-85.

Bierbrauer, F. (2004). Mathematical modeling of water-droplet impact on hot galvanized steel surfaces. (PhD, School of Mathematics and Applied Statistics, University of Wollongong).

Bonometti, T., & Magnaudet, J. (2007). An interface-capturing method for incompressible two-phase flows. validation and application to bubble dynamics. *International Journal of Multiphase Flow, 33*(2), 109-133.

Brackbill, J. U., Kothe, D. B., & Zemach, C. (1992). A continuum method for modeling surface tension. *Journal of Computational Physics, 100*(2), 335-354.

Built-in parallel computing support in MathWorks products. Retrieved May-29, 2010, from http://www.mathworks.com/products/parallel-computing/builtin-parallel-support.html

Chen, S., He, X., & Luo, L. -. (2007). Lattice Boltzmann models for multiphase flows. In A. Prosperetti, & G. Tryggvason (Eds.), *Computational methods for multiphase flow* (pp. 157-192). Cambridge: Cambridge University Press.

Chen, S., Johnson, D. B., & Raad, P. E. (1991). The surface marker method. In L. C. Eroble, & C. A. Brebbia (Eds.), *Computational modeling of free and moving boundary problems, vol. 1, fluid flow* (pp. 223-234). New York: de Gruyter.

Chen, S., Johnson, D. B., Raad, P. E., & Fadda, D. (1997). The surface marker and micro cell method. *International Journal for Numerical Methods in Fluids, 25*(7), 749-778.

Enright, D., Fedkiw, R., Ferziger, J., & Mitchell, I. (2002). A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics, 183*(1), 83-116.

Ferziger, J. H., & Peric, M. (2002). *Computational methods for fluid dynamics* (3rd ed.). Berline: Springer Verlag.

Floryan, J. M., & Rasmussen, H. (1989). Numerical methods for viscous flows with moving boundaries. *Applied Mechanics Reviews, 42*(12), 323-341.

Gao, D., Morley, N. B., & Dhir, V. (2003). Numerical simulation of wavy falling film flow using VOF method. *Journal of Computational Physics, 192*(2), 624-642.

Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Royal Astronomical Society, Monthly Notices, 181*, 375-389.

Harlow, F. H. (1955). *A machine calculation method for hydrodynamic problems* (Tech. Rep No. LAMS-1956). Los Alamos: Los Alamos National Laboratory Report.

Harlow, F. H., & Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids, 8*(12), 2182-2189.

Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics, 39*(1), 201-225.

Hua, J., Stene, J. F., & Lin, P. (2008). Numerical simulation of 3D bubbles rising in viscous liquids using a front tracking method. *J.Comput.Phys., 227*(6), 3358-3382.

Johnson, N. L. (1996). The legacy and future of CFD at Los Alamos. *Proceedings of the 1996 Canadian CFD Conference,* Ottawa, Canada.

Kapitza, P. L., & Kapitza, S. P. (1964). Collected papers of P.L. kapitza. In D. t. Haar (Ed.), *Wave flow in thin layers of a viscous fluid* (pp. 669-709). New York: The Macmillan Company.

Kim, S. H., & Pitch, H. (2009). On the lattice Boltzmann method for multiphase flows. *Annual Research Briefs, Center for Turbulence Research, 2009*, 377.

Kothe, D. B., Mjolsness, R. C., & Torrey, M. D. (1991). *RIPPLE: A computer program for incompressible flows with free surfaces* (Tech. Rep No. LA-12007-MS). Los Alamos: Los Alamos National Laboratory.

Kothe, D. B. (1999). Perspective on Eulerian finite volume methods for incompressible interfacial flows. *Free surface flows* (pp. 267-331). New York: Springer Verlag.

Kunugi, T., & Kino, C. (2005). DNS of falling film structure and heat transfer via MARS method. *Computers & Structures, 83*(6-7), 455-462.

Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., & Zanetti, G. (1994). Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics, 113*(1), 134-147.

LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*. Cambridge; New York: Cambridge University Press.

Li, C., Garimella, S. V., Reizes, j. A., & Leonardi, E. (1999). The development of a bubble rising in a viscous liquid. *Journal of Fluid Mechanics, 387*, 61.

Li, S., & Liu, W. K. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews, 55*(1), 1-34.

Liu, J., & Gollub, J. P. (1994). Solitary wave dynamics of film flows. *Physics of Fluids, 6*(5), 1702-1712.

Loubenets, A. (2007). An immersed finite element method and its application to multiphase problems. (PhD, Kungliga Tekniska högskolan).

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *Astronomical Journal, 82*, 1013-1024.

McNamara, G. R., & Zanetti, G. (1988). Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters, 61*, 2332.

Nichols, B. D., Hirt, C. W., & Hotchkiss, R. S. (1980). *SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries* (Tech. Rep No. LA-8355). Los Alamos: Los Alamos National Laboratory.

Noh, W. F., & Woodward, P. (1976). (SLIC) simple line interface calculation. In A. L. van de Vooreb, & P. J. Zandbergen (Eds.), *Lecture notes in physics* (pp. 330-340). New York: Springer-Verlag.

Nosoko, T., Yoshimura, P. N., Nagata, T., & Oyakawa, K. (1996). Characteristics of two-dimensional waves on a falling liquid film. *Chemical Engineering Science, 51*(5), 725-732.

Olsson, E., & Kreiss, G. (2005). A conservative level set method for two phase flow. *Journal of Computational Physics, 210*(1), 225-246.

Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics, 79*(1), 12-49.

Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of Computational Physics, 25*(3), 220-252.

Pilliod, J. E., & Puckett, E. G. (2004). Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics, 199*(2), 465-502.

Popinet, S., & Zaleski, S. (1999). A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids, 30*(6), 775-793.

Prosperetti, A., & Tryggvason, G. (2007). A computational approach to multiphase flow. In A. Prosperetti, & G. Tryggvason (Eds.), *Computational methods for multiphase flow* (pp. 1-18). Cambridge: Cambridge University Press.

Puckett, E. G., Almgren, A. S., Bell, J. B., Marcus, D. L., & Rider, W. J. (1997). A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics, 130*(2), 269-282.

Renardy, Y. Y., Renardy, M., & Cristini, V. (2002). A new volume-of-fluid formulation for surfactants and simulations of drop deformation under shear at a low viscosity ratio. *European Journal of Mechanics - B/Fluids, 21*(1), 49-59.

Rider, W. J., & Kothe, D. B. (1998). Reconstructing volume tracking, *Journal of Computational Physics, 141*(2), 112-152.

Rudman, M. (1997). Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids, 24*(7), 671-691.

Salih, A., & Ghosh Moulic, S. (2009). Some numerical studies of interface advection properties of level set method. *Sadhana, 34*(2), 271-298.

Scardovelli, R., & Zaleski, S. (2003). Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *International Journal for Numerical Methods in Fluids, 41*(3), 251-274.

Scardovelli, R., & Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech., 31*, 567-603.

Shyy, W., Francois, M., & Udaykumar, H. S. (2001). 10 Cartesian and curvilinear grid methods for multi-domain, moving boundary problems. *Thirteenth International Conference on Domain Decomposition Methods,* Lyon, France.

Sussman, M., Smith, K. M., Hussaini, M. Y., Ohta, M., & Zhi-Wei, R. (2007). A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics, 221*(2), 469-505.

Sussman, M., & Puckett, E. G. (2000). A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics, 162*(2), 301-337.

Sweby, P. K. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis, 21*(5), 995-1011.

Tryggvason, G., & Balachandar, S. (2007). Direct numerical simulations of finite reynolds number flows. In A. Prosperetti, & G. Tryggvason (Eds.), *Computational methods for multiphase flow* (pp. 19-36). Cambridge: Cambridge University Press.

Tryggvason, G., Bunner, B., Ebrat, O., & Tauber, W. (1998). Computations of multiphase flows by a finite difference/front tracking method. I. multi-fluid flows. *VKI lecture series 1998-03*. Rhode-St-Genèse, Belgium: The von Karman Institute for Fluid Dynamics.

Tryggvason, G., Sussman, M., & Hussaini, M. Y. (2007). Immersed boundary methods for fluid interfaces. In A. Prosperetti, & G. Tryggvason (Eds.), *Computational methods for multiphase flow* (pp. 37-77). Cambridge: Cambridge University Press.

Unverdi, S. O., & Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics, 100*(1), 25-37.

van Sint Annaland, M., Deen, N. G., & Kuipers, J. A. M. (2005). Numerical simulation of gas–liquid–solid flows using a combined front tracking and discrete particle method. *Chemical Engineering Science, 60*(22), 6188-6198.

Versteeg, H., & Malalasekra, W. (2007). *An introduction to computational fluid dynamics: The finite volume method* (2nd ed.). Harlow: Prentice Hall.

Yeoh, G. H., & Tu, J. (2010). *Computational techniques for multiphase flows*. Oxford: Butterworth-Heinemann.

Youngs, D. L. (1982). Time-dependent multi-material flow with large fluid distortion. In K. W. Morton, & M. J. Baines (Eds.), *Numerical methods for fluids dynamics* (pp. 273-285). New York: Academic Press.

Zijlema, K., & Wesseling, P. (1995). *Higher order flux-limiting methods for steady-state, multidimensional, convection-dominated flow* (Tech. Rep No. 95-131). Delft, Netherlands: Faculty of technical mathematics and informatics, Delft university of technology.